

A Hassle-free System for Ordering Websites: twopointz3ro (Prototype)



Nishio, Eric

Laurea University of Applied Sciences
Laurea Leppävaara

**A Hassle-free System for Ordering Websites:
twopointz3ro (Prototype)**

Nishio, Eric
Business Information Technology
Thesis
July 2010

Nishio, Eric

A Hassle-free System for Ordering Websites: twopointz3ro (Prototype)

Year	2010	Pages	30
------	------	-------	----

A hassle-free system for ordering websites is an example of the modern Web 2.0 trend that has emerged in the second half of the past decade; opting for simplicity and interconnectedness. The mainstream population of the Internet is demanding easier means to establishing an online presence—in the form of a dot com website or equivalent. This project, therefore, focuses on designing and developing a simple and user-friendly system for purchasing a fully functional website that does not require technical expertise.

Many service providers on the Web have failed to identify the demand to expand their awareness by tuning out the needs of the mainstream, and not realizing the fact that website ownership is no longer a phenomenon only involving computer programmers and major corporations. So in order to accommodate newcomers who desire to have an extension to their Facebook and Twitter profiles, web development services must be updated and brought to these groups by making the acquisition process as clear and easy-to-understand as possible. The core objective of this project, therefore, is to investigate the requirements of the general Internet user, and create a system that allows anyone to comfortably order a suitable website for his or her purposes, whether it be business-related or exclusively personal.

The methods used during the research phase consist of benchmarking and interviewing, whereas the development phase was governed by web development methods such as HTML, CSS, MySQL, and PHP, as well as Adobe Illustrator and Photoshop for graphic design. Benchmarks were conducted on various websites and e-commerce sites for critically evaluating their user-friendliness and for extracting knowledge on how to develop more efficient websites. One project manager was also interviewed in search of advice that would shed light on how to improve web development services.

The research provided practical insights on how to adopt a minimalist approach to web design and service by eliminating distractions that cause the customer's attention to veer away from the main objective—the acquisition of a website. This mindset demanded that the system be optimized for its key purpose by carefully handpicking what to include and what to discard.

In conclusion, the project was a success, both in terms of fulfilling the objective of developing a user-friendly ordering system for mainstream Web users, and providing invaluable personal knowledge and experience related to web development.

Keywords website, ordering, PHP, MySQL, web design, Web 2.0, user-friendly

Nishio, Eric

Vaivaton järjestelmä nettisivujen tilaamiseen: twopointz3ro (Proto)

Vuosi 2010 Sivujen lukumäärä 30

Vaivaton järjestelmä nettisivujen tilaamiseen on esimerkkisovellus viime vuosikymmenen loppupuolella ilmestyneestä Web 2.0 -ilmiöstä, jonka lähtökohtina ovat muun muassa yksinkertaisuus ja yhtenäisyys. Internetin väestö näin ollen toivookin helpompia mahdollisuuksia tuoda itsensä verkkoon ”dot com” -sivuston tai vastaavan profiilin muodossa. Tämä työ keskittyy yksinkertaisemman ja käyttäjäystävällisemmän tilausjärjestelmän suunnitteluun ja kehitykseen, jonka avulla kaikki pääsisivät laatimaan omat verkkosivut ilman teknistä osaamista.

Useat yritykset ja palveluntarjoajat ovat epäonnistuneet huomioimaan tarpeen laajentaa vaikutusvaltaansa sulkemalla kasvavan Internetin väestön potentiaalin asiakkuuteen, sillä verkkosivujen omistaminen ole enää rajattu ohjelmoijiin ja suurorganisaatioihin. Jotta uusille tulokkaille voitaisiin tarjota samoja palveluita esimerkiksi integroimalla Facebook- tai Twitter-profiilit verkkosivuihin, tulisi web-kehityspalvelut päivittää ja tuotteiden saatavuus mahdollistaa kaikille. Tämän työn päätarkoitus onkin näin ollen tutkia peruskäyttäjien tarpeita ja luoda järjestelmä, joka antaa kenen tahansa laatia ongelmitta verkkosivut vastaamaan tarpeitaan.

Työssä tutkimukseen käytetyt menetelmät koostuvat benchmarkauksesta ja haastattelusta, kun taas itse järjestelmän kehityksessä käytettiin HTML-, CSS-, MySQL- ja PHP-menetelmiä sekä Adobe Illustratoria ja Photoshopia graafiseen suunnitteluun. Benchmarkaus eli analysointi mahdollisti muiden verkkosivujen evaluoinnin käyttäjäystävällisyydessä ja tiedon hankkimisessa tehokkaampien verkkosivujen suunnitteluun liittyen. Projektipäällikön haastatteleminen toi neuvoja verkkosivupalvelujen parantamiseen.

Alustava tutkimus toi käytännönläheistä ymmärrystä minimalistisen lähestymistavan käyttöönotosta web-suunnittelussa ja -palveluntarjonnassa poistamalla häiriötekijät, jotka muuten suistaisivat asiakkaan keskittymisen päämäärästä eli verkkosivujen vaivattomasta hankkimisesta. Tämä lähestymistapa vaati järjestelmän optimointia päätarkoitustaan varten siten, että tarpeet tuli valita erittäin tarkasti poistamalla kaikki tarpeeton.

Kaiken kaikkiaan työ oli onnistunut sekä opinnäytetyön päämäärän täyttämässä että käyttäjäystävällisen tilausjärjestelmän kehityksessä. Projekti antoi myös runsaasti korvaamatonta tietoa ja kokemusta web-suunnitteluun ja -kehitykseen liittyen.

Contents

1	Introduction	7
1.1	How the Idea Got Started.....	7
1.2	Demands for Development.....	7
1.3	Target Market.....	8
2	Objective	8
3	Web-based Ordering System.....	9
4	Methods, Tools, and Techniques.....	10
4.1	Benchmarking.....	10
4.2	Interview	11
4.3	Adobe Photoshop and Illustrator.....	11
4.4	HTML and CSS.....	11
4.5	MySQL.....	12
4.6	PHP	12
4.7	UML.....	13
5	Requirements Analysis.....	13
5.1	Appropriate Language	13
5.2	Minimalist Design.....	14
5.3	Intuitiveness	15
5.4	Customizability.....	15
6	Development.....	16
6.1	Modeling.....	16
6.1.1	Use-Case Diagram.....	16
6.1.2	Database Diagram.....	18
6.2	Visual Design.....	22
6.2.1	Structure	22
6.2.2	Graphics.....	23
7	Demonstration	25
7.1	Ordering a Pre-Compiled Package.....	25
7.2	Ordering a Customized Package	30
8	Project Phases	34
9	Further Development.....	35
10	Conclusion	36
	References	37
	Figures.....	38
	Attachment 1 Benchmarking	39
	Attachment 2 Interview.....	44
	Attachment 3 HTML and CSS	46
	Attachment 4 MySQL Database	52

Attachment 5 PHP Scripts54

1 Introduction

This project, titled *twopointz3ro*, is an attempt to create a clear and user-friendly system for placing website orders. The main goal is to let customers with little technical knowledge of computing to purchase a website solution that suits their purposes—whether it be related to business, hobbies, or one’s personal life.

The key principles applied to this approach are simplicity, intuitiveness, and clear customizability. When the user interface is uncluttered and free of distractions, it allows the customer to focus his or her attention on finding the most appropriate website solution. An intuitive layout of the ordering process reduces the time spent looking for the right options: “Noticeability is in the hands of the designer, and it’s the designer’s job to make sure that the right things are noticed first.” (Hunt 2008, 74).

With clear customizability, the system provides the customer with an array of convenient choices and form fields for acquiring the most suitable website solution to accommodate his or her needs. A cohesive relationship between the three aforementioned principles allows the customer to place an order with minimum hassle.

1.1 How the Idea Got Started

The reason why I decided to develop this system is that many web design services require their customers to have a fairly sophisticated understanding of how websites work. Fewer distractions will let customers get what they are looking for in a comfortable and intuitive way.

I am interested in starting my own business in the web development niche, which is also why I am pursuing an entrepreneurial path instead of fulfilling the needs of a single, exclusive customer—that is, the employer. I believe that by creating this system, many customers will be able to benefit from the convenience of a simple and user-friendly website purchasing system.

1.2 Demands for Development

The Internet is in a state of constant change. The writers of *Web 2.0: New Tools, New Schools* predict “technology will literally transform every aspect of business, every aspect of life, and every aspect of society” (Solomon & Schrum 2007, 10). Web-based services are constantly undergoing a major change, as more and more people are getting online. Service providers

should understand the fact that in the year 2010, most of the Internet population is not *tech-savvy*, that is, technologically . While the general public is becoming more computer-literate than ten years ago, and even more so thanks to the social media boom, most people still do not possess the technical know-how required to manage, or let alone purchase an appropriate website for their needs.

If the computer experts want to upgrade their services for the new computer era, they must understand the necessity to shift toward simplicity and user-friendliness. Offline businesses—that is, companies that do not operate on the Internet—are also slowly grasping the need to expand their operations into the virtual world. Aspiring individuals, freelancers and entrepreneurs alike will also need to be catered to, and the only way to enable everyone to have an online presence is to provide the simplest possible method for acquiring a website that satisfies the needs of its buyer.

1.3 Target Market

This project primarily targets freelancers, entrepreneurs, and small-sized businesses. The aim is to provide effective website solutions for minor businesses, and to possibly create a long-term relationship with them by offering consulting services and various add-ons that can supplement websites already in existence.

2 Objective

The objective of this project is to create a website purchasing system that will let companies and individuals buy websites with as little inconvenience as possible. Websites typically employ a number of technical concepts, making it difficult for the mainstream of Internet users to understand concepts along the lines of *domain*, *HTML*, *cgi-bin*, or *MySQL database*.

In the recent years, a new concept has emerged within web development circles called *Web 2.0*. This “new version” of the Web is more of a philosophy subscribing to the modern beliefs of many leading web designers and developers that websites should be made simpler, clearer, and easier to learn to use. This does not mean that online-goers are becoming dumber every year; it only means that the Internet is appealing to a constantly increasing number of people. The Internet is no longer an exclusive place inhabited by computer programmers and online gamers. It has become a community housing a population of gardeners, cooks, housewives, school children, soldiers, and pilots. This is why it is important to allow this new population of Internet users to establish an online presence comfortably, without the need to know what it means to upload a hypertext file onto the server or whether one should get a website supporting either PHP or ASP. Readers (and customers) should not be required to

think how something works: “[‘Don’t make me think!’] is the overriding principle—the ultimate tie breaker when deciding whether something works or doesn’t in a Web design” (Krug 2005, 11). The customer does not necessarily need to know what happens on the server as long as the website itself satisfies the customer’s needs.

I personally believe in a minimalistic approach to product marketing, which in practice entails the simplification of concepts for the customer so that he will not feel bewildered by a vocabulary of foreign-sounding words. If a customer is not familiar with technical concepts, he or she will have problems deciding what to purchase. This is why it is preferable that a more effective way of ordering websites be developed, which would accommodate a wider range of customers.

The primary job of the service is to build customized websites for companies and individuals. But it would additionally offer other necessary services as well, such as consulting and technical maintenance. For some customers it may suffice to be provided only with a mere website layout, so there will also be an option to order a graphical design. Others, however, may need technical consulting to push their online presence into motion. Concepts such as the social media are certainly not self-explanatory for every entrepreneur and businessperson alike. This type of consulting would range from offering simple technical advice to teaching the customer how to market her services using ads on Facebook.

3 Web-based Ordering System

A web-based ordering system is, in simple terms, a website where customers can place orders. These ordering systems can be seen in many places on the Web; every website that has a page where you can type in your name and address to buy a product is a web-based ordering system. eBay.com, Gucci.com, Amazon.com, and any other online store have got one.

In this project, the emphasis is on ordering websites. While twopointz3ro does offer pre-compiled website packages, a customized package is, after all, quite an abstract product to purchase. If the customer wishes to buy a customized website, she must define the product herself. Of course there are certain constraints to customizing a website, since it is rather impossible to cater to all the technologies in the world, but the path of customization means that the customer can put all the individual components in the shopping cart, chooses the colors and visual tones for the graphical layout, and makes a rough plan of what will be displayed on each page. But the idea in this project is to make customizations as easy and intuitive as possible. There will be a minimum of technical terminology. Some words such as *social media*, *mailing list*, or *PayPal* cannot be omitted due to the fact that some customers

are particularly interested in incorporating those technologies into their website. Still, the website should not use confusing terms, such as the very technologies that the site itself was built with (chiefly PHP, HTML, and CSS).

A website that sells products usually has a selection of payment options for the customer. This project does provide future support for making instant payments on the website, but does not implement it yet because the business plan is still in development, and this project is not intended to be launched immediately upon completion of the ordering system. There is, however, always the option of requesting payments via email, so the managing of payments is not a practical constraint, nor is it a deficiency marking it dysfunctional as a service.

4 Methods, Tools, and Techniques

The ordering system is fundamentally a website, so it requires a graphical layout, a database, and scripts. Adobe Photoshop and Illustrator were chosen for creating the cosmetic appearance of the graphical interface, HTML/CSS for structuring the layout, MySQL for the database, and PHP for programming the scripts to communicate with the database. In addition, UML (Unified Modeling Language) was used to plan the system visually. Other methods, such as interviews and benchmarks, were also used for analyzing the requirements of the project.

4.1 Benchmarking

To gain better insight into the usability and effectiveness of a web-based ordering system, an analysis was conducted. The benchmark analyzed a number of different e-commerce websites that specialize in selling products to customers, and evaluated their strengths and weaknesses. The underlying purpose for gathering this data was to make the ordering system as effective as possible.

The aim of the project is to develop the best interface and functionality in the market for a web design provider, so the analyses naturally focused on collecting useful data only from other successful and highly acclaimed websites and companies. Some of the websites included in this benchmark have been rated as examples of well-designed e-commerce websites by web design company *spyrestudios* (Snell 2010).

The analysis is not limited to websites of other web service providers. Analyses were conducted on a variety of types of websites, ranging from restaurants to clothing stores. A diverse test group enables us to draw inspiration from unrelated e-commerce websites, providing new ideas that lead to innovative solutions.

4.2 Interview

I conducted an interview where I talked with a project manager at Laurea University of Applied Sciences who was a customer when purchasing a website for a project. She had no prior experience with websites, so she qualified extremely well as an interviewee. The purpose was to extract valuable knowledge on what customers tend to look for in commercial website packages, and to discover which of the services would facilitate the process of bringing their business online.

I looked at different problems and needs from the standpoint of individual people potentially looking for websites for their business. The goal was to find out which aspects should specifically be made simpler and easier to use, and which could be omitted altogether. This allows us to simplify the system to the greatest possible extent, and discover the core mechanics needed for a successful website solution.

4.3 Adobe Photoshop and Illustrator

Adobe's Creative Suite package includes the essentials for creating professional-looking web designs for any purpose. This project, however, only requires the help of Photoshop and Illustrator, because the website follows a minimalist approach to design, therefore eliminating the need for Macromedia Flash (animated graphics), video, and audio. Both software can be used for designing the visual layout of the website.

Most of the images are done in Photoshop, because it provides the developer with a wide array of tools for web publishing. Photoshop is particularly effective for slicing elements, tweaking colors and various other cosmetic qualities.

Illustrator is chiefly a vector tool, which is why it was used it for creating the logos and other elements that may need to be resized at some point. Vector images do not lose their quality when resized, so Illustrator is the optimal tool for drawing the iconic components of the website.

4.4 HTML and CSS

Websites are basically created with a simple markup language called HTML, which appropriately stands for 'Hypertext Markup Language'. The structure of every single web page in this project is underpinned by HTML. Without HTML, nothing would be visible. The primary

role of HTML, therefore, is to display the images and text in a structured way so that the viewers will be able to locate the information they are looking for.

Cascading Style Sheets, or CSS in short, is a markup language for the effective stylization of the elements displayed on a web page. It is particularly efficient in its way of synchronizing and connecting formatting styles throughout all the pages of the website. The cosmetic properties included in a single style sheet can easily be applied to hundreds of individual pages without having to modify each page individually.

The website in this project is, therefore, structured in HTML, and the site elements synchronously defined with CSS. All in all, the visual design itself consists of elements created with Photoshop, Illustrator, HTML, and CSS.

4.5 MySQL

MySQL is an open-source database software based on SQL (Structured Query Language) that is used for managing information in relational database management systems. It is a widely used database system, partly because of its open source philosophy, which makes it convenient for developers to find comprehensive documentation on it.

This project requires a database to store all the data in, so MySQL serves as the technology for maintaining information in a database. The key reason for choosing MySQL over other database software is also affected by MySQL's integral connection with PHP. And both MySQL and PHP are virtually OS-independent: "One of the best features of both PHP and MySQL is that they work with any major operating system and many of the minor ones" (Welling & Thomson 2009, 3).

4.6 PHP

PHP: Hypertext Processor (often just referred to as 'PHP') is a scripting language nowadays primarily used for developing dynamic websites. PHP is also an excellent tool for connecting to a MySQL database and managing information within it. The language includes a rich collection of useful functions that can be directly applied to web pages or used as components in one's own scripts and functions.

The purpose of PHP in this project is to work in conjunction with the MySQL database and HTML. The website adds and retrieves information to and from the database using scripts programmed in PHP.

4.7 UML

Unified Modeling Language, abbreviated to 'UML', is a standardized modeling language used to visually plan computer software. By creating a comprehensive map of the software project at hand, the developer gains access to a visual reference that can be used to implement the components of the project.

This project uses UML's component diagrams to plan the website's functions from both the customer's perspective as well as the manager's, so that even outsiders will be able to understand the mechanics of how the system is meant to work.

The MySQL database will also be first modeled with UML class diagrams to make the implementation process as clear and well structured as possible. By having a comprehensive set of UML diagrams for the database tables involved, it will also be less burdensome to plan changes to the database.

The UML modeling applications used in this project are ArgoUML and NetBeans for Mac OS X. The screen captures of the models in this report are images that display how the diagrams are laid out within ArgoUML and NetBeans.

5 Requirements Analysis

There are four principal areas that require special attention when creating an online ordering system that is geared toward the general public: appropriate language, minimalist design, intuitiveness, and customizability. The following sections will discuss how the aforementioned attributes can be optimized for this project.

The benchmark conducted on an array of existing websites also serves as an underpinning guide which helps to develop a more optimized system that satisfies the demands of the current generation of customers. The requirements analysis will, therefore, be based on the benchmark to a significant extent.

5.1 Appropriate Language

When the target customer is a person who does not have a strong background in computer science, good communication is paramount in securing the customer. If the website uses too much technical jargon, the reader will feel alienated and will probably seek alternate means to acquire a website. But when the website establishes a connection between itself and the

customer by talking conversationally to her in *her* language, she will probably feel more inclined to stay with that provider because everything is clear and understandable.

The key principle in establishing that kind of relationship with the customer is to omit as many technical words as possible, and preferably replace them with words that are intelligible for the general reader—that is, someone with no affiliation to the technical side of computers. The most effective practice, therefore, is to employ common words. This practice does not exclusively apply to technical words; it is also important to select short and succinct words that are *pure* English. Longer loanwords that derive from Latin should also be avoided in order to create unnecessary confusion.

A conversational voice works best. The “six-figure blogger” Darren Rowse (who started off as a blog hobbyist but is now earning a lavish income from his blogs) states in one of his blog posts on the marketing of blogs that “One of the things that can help [to gain loyal readers] is to simply write in a relational or conversational style.” (Rowse 2010) For example, it is obvious that “What do you need?” is much easier to grasp for the general reader than a stiff phrase along the lines of “Define your requirements.” There is really no need to choose the “more formal” phrase when the same can be expressed in simpler terms. Especially when the target customer is someone with little technical know-how, it would be like leaving money on the table if the company lost customer because they made the regrettable choice of making their website too difficult to understand. Fancy words do not make a website look professional—usability does.

5.2 Minimalist Design

When it comes to effective usability, a clean layout is the cornerstone without which it is impossible to maintain a website where people can easily find what they are looking for. Looks are important.

This, however, certainly does not mean that a successful website needs to have all the bells and whistles that make it look like carnival poster. When it comes to websites, a clean design naturally communicates better usability than a cluttered page where customers have to try hard to orient themselves.

The best way, therefore, is to adopt a minimalist style of design that attempts to eliminate all unnecessary things, and focuses on the essential task of ordering a website. This means that the twopointz3ro website itself will consist of a header with a logo, a footer displaying contact information, and the main body where the essentials will be placed. There will be no

sidebars, because they would only distract the reader from the main path. To maintain conciseness, the number of words will also be kept to a minimum.

5.3 Intuitiveness

In order to optimize usability, the design must be coupled with an intuitive interface. Intuitiveness refers to the learning curve of a particular system. This project deals with an ordering system, so it is important that all customers are able to figure out where to start the process and how to complete the order.

For a tool to be functional, its user has to learn how to use it. Good websites should not require any learning, because they have been compelled to accommodate visitors. Design conventions (for example, the placement of the logo and menu bar) are important for guiding readers toward the right direction, but it is also wise to consider which elements are necessary and where they should belong.

In order to place a valid order, the customer needs to choose or customize his product, input his personal information, and submit the order. To ease the process, it is ideal to have as few individual pages or sections as possible. One page should also house a minimal number of form fields to fill. This is why the minimalist approach works best in conjunction with intuitiveness, because the absence of clutter makes it much less burdensome to locate the right things, enter the right information, and finally log off and complete the process.

5.4 Customizability

Customizability refers to service whereby customers can tailor their purchases to accommodate her needs. When we are dealing with website solutions, there are many choices, and it is impossible to satisfy everyone's needs. This project's main target group is the mainstream that populates the Internet, which is why it is not as difficult to narrow down the options as it would be when a customer is looking for a wide array of technical solutions that need licenses and other financially challenging prerequisites.

So to accommodate the general citizen of the Internet, service providers must know where to place emphasis. They need to know what the general public wants and how they should deliver it to them. An open contact form, where they can make specific requests written in plain English, can be effective to get a certain kind of message across to the developers, but its open nature defeats the purpose because all customers do not necessarily know what it really is that they want. So it is the service provider's job to guide them to the right

direction—first by providing a basic tool that inquires customers about their most important needs.

This tool, which asks the customer what she wants, is delivered to the customer in the form of an online form that needs to be filled out as accurately as possible. The customer is given the freedom to input whatever she likes, but the effectiveness of the form comes from the simple questions asked. This form establishes a relationship with the customer while allowing there to be changes made to the product when the need arises.

6 Development

The following accounts describe the steps taken to developing the final work. All the code, however, has been appended in the attachments at the end of this document.

6.1 Modeling

The initial modeling phase of the project focused on constructing a rough skeleton based on the points discovered during the requirements analysis. The main reason the system was modeled in UML was to gain a visual clue on how to build the system: “Modeling helps you see the forest for the trees, allowing you to focus on, capture, document, and communicate the important aspects of your system’s design” (Miles & Hamilton 2006, 1). As this report will reveal, only the models used in this project were the *rough* blueprints that functioned, in the end, as an approximate framework. Some items were, therefore, omitted in the final work due to practical reasons that did not necessitate their use.

6.1.1 Use-Case Diagram

The first of the two models used in the project is the use-case diagram (Figure 1), which depicts the hypothetical steps that a customer would take when ordering a product via the system.

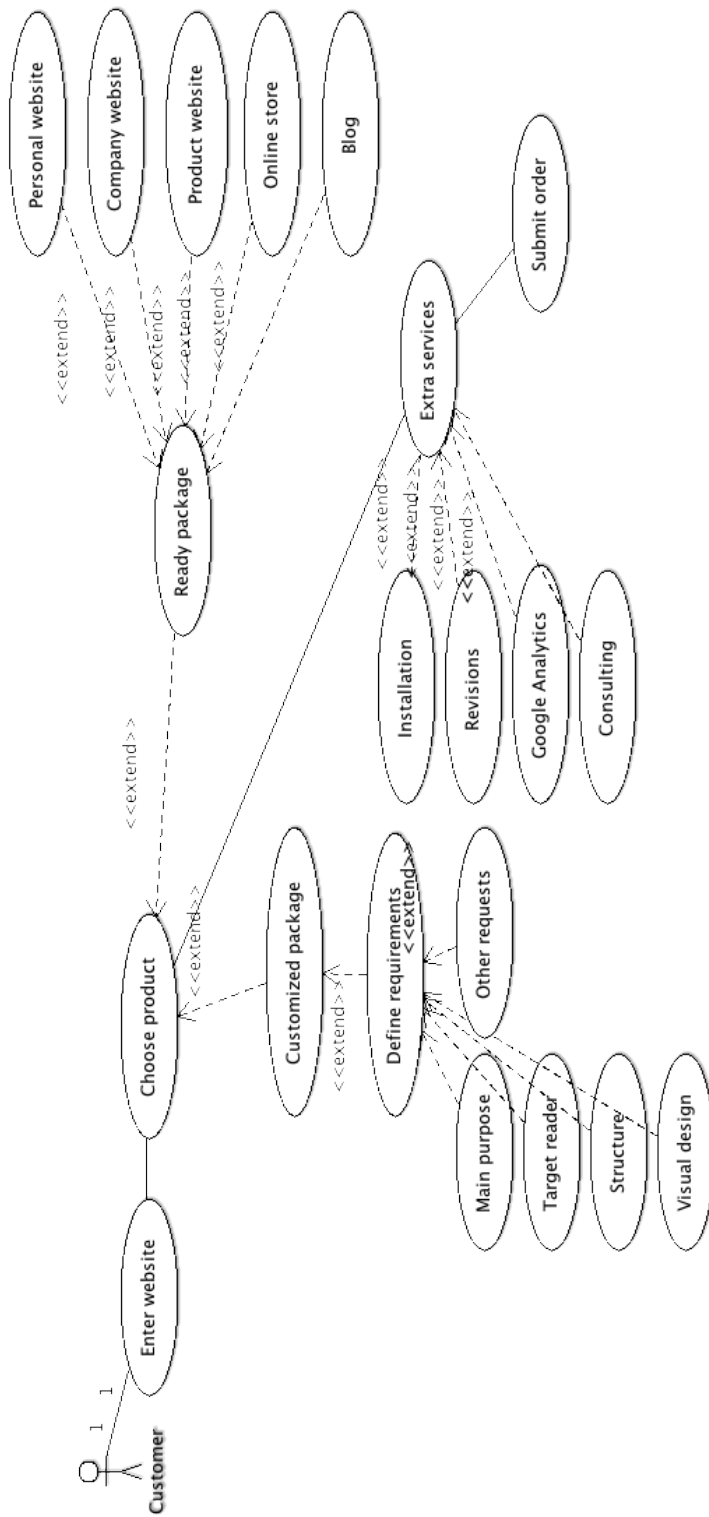


Figure 1: Use-Case Diagram

In the use-case diagram, the focus is on the customer, who first enters the website's main starting page, selects either a pre-compiled website package or initiates the customization tool. This is the point where the path changes.

The simple pre-compiled package only displays a fixed number of ready packages (personal website, company website, product website, online store, or blog) available for purchase. There is no customization option available for these products, which means that the next step merely deals with placing the order by inputting one's user information into the form. The data transmitted to the database via this form is then read by the developer and used as a blueprint for creating a website for the customer.

The customization option allows the customer to use the online tool to give specific information on how he wants the website to look, feel, and work. The key here is to give detailed information on the fundamental qualities of the website, such as colors, tones, topic, target readers, purpose, and so forth, as well as additional components such as Google Analytics, discussion forums, contact forms, and shopping carts. This step establishes the relationship between the customer and the developer, first by defining the key elements of the website, and by allowing the developer to contact the customer via email in order to discuss the qualities and attributes in greater detail if needed.

6.1.2 Database Diagram

The database diagram represents the MySQL database where all customer information and order data are stored. The UML diagram shows three database tables, but the *Package* table was omitted because it was not seen as necessary to keeping the website organized, since the products themselves will not be constantly modified. The final decision, therefore, was to focus on creating two kinds of database tables for storing data exclusively related to customers' orders: that is, information about the customer and his order. The orders would be further split into two separate tables: pre-compiled product orders and customized orders.

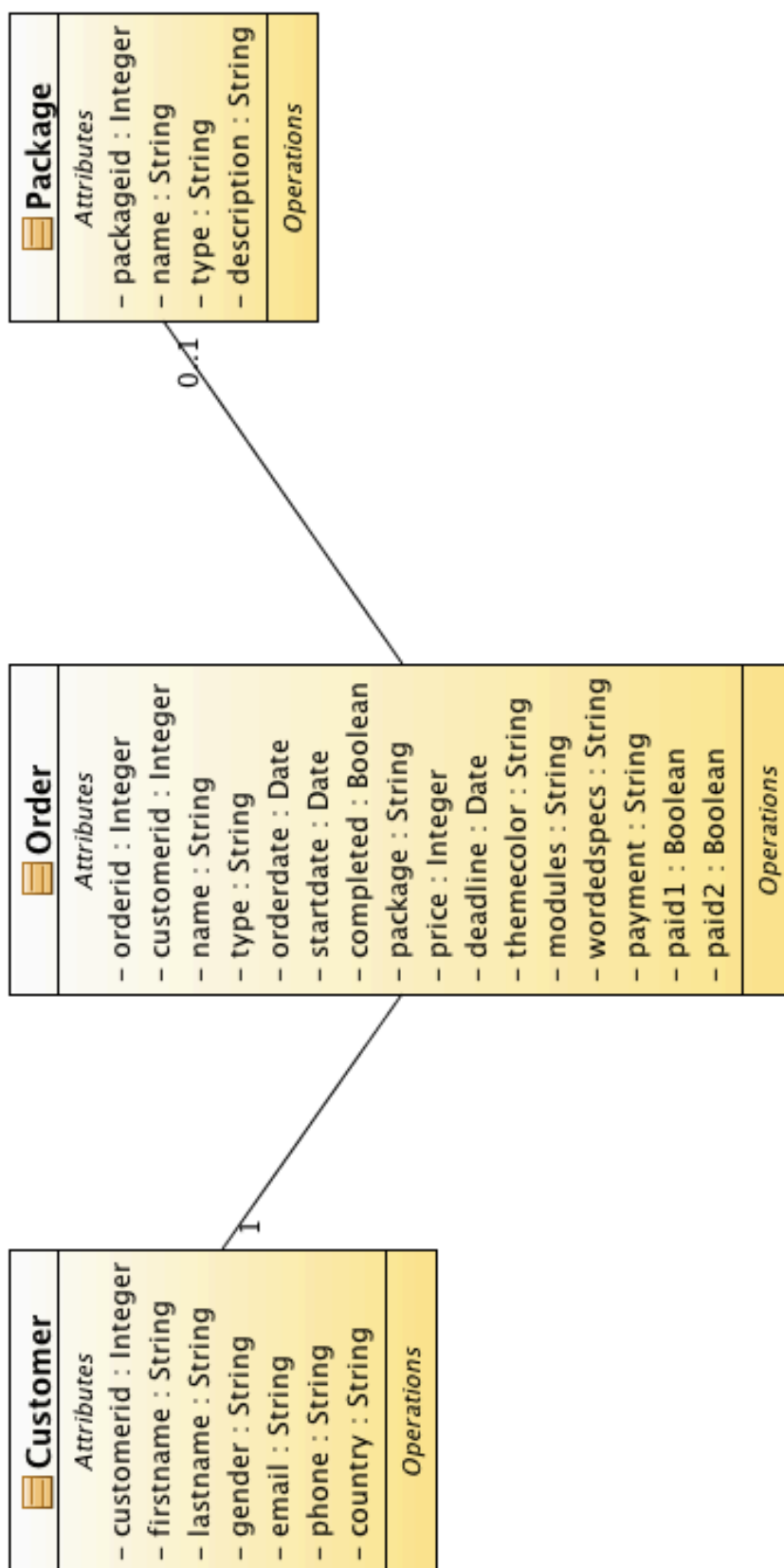


Figure 2: Original Database Diagram

The *customer* table (Figure 3) is the part of the database that houses information on every individual customer who has placed an order through the system. The table includes a total of 12 fields: a unique (primary key), numeric customer ID (*customerid*) for each customer to clearly distinguish between all the customers; first name; last name; gender; two address fields; postal code/ZIP; city; country; phone number; email address; and password. The key fields here are *email* and *password*, as they can be used by the customer to place orders in the future without the burden of registering again. The email address will also be used by the developer to communicate with the customer. The customer ID field is of course used by the system in conjunction with the order in order to identify the “owner” of an order.





















































































	Field	Type	Collation	Attributes	Null	Default	Extra	Action
<input type="checkbox"/>	customerid	int(10)			No	None	auto_increment	      
<input type="checkbox"/>	firstname	varchar(15)	latin1_swedish_ci		No	None		      
<input type="checkbox"/>	lastname	varchar(30)	latin1_swedish_ci		No	None		      
<input type="checkbox"/>	gender	varchar(6)	latin1_swedish_ci		No	None		      
<input type="checkbox"/>	address1	varchar(30)	latin1_swedish_ci		No	None		      
<input type="checkbox"/>	address2	varchar(30)	latin1_swedish_ci		No	None		      
<input type="checkbox"/>	postalcode	int(15)			No	None		      
<input type="checkbox"/>	city	varchar(25)	latin1_swedish_ci		No	None		      
<input type="checkbox"/>	phonenumber	varchar(40)	latin1_swedish_ci		No	None		      
<input type="checkbox"/>	email	varchar(50)	latin1_swedish_ci		No	None		      
<input type="checkbox"/>	country	varchar(20)	latin1_swedish_ci		No	None		      
<input type="checkbox"/>	password	varchar(20)	latin1_swedish_ci		No	None		      

Figure 3: Final Customer Table in phpMyAdmin

The pre-compiled order table, *premadepackorder* (Figure 4), does not need many fields because it only needs to display which pre-compiled product has been ordered, by whom, and at what price. Every individual order should also have a unique, numerical ID so that the developer can clearly identify it.





























	Field	Type	Collation	Attributes	Null	Default	Extra	Action
<input type="checkbox"/>	premadepackorderid	int(15)			No	None	auto_increment	      
<input type="checkbox"/>	customerid	int(10)			No	None		      
<input type="checkbox"/>	productname	varchar(50)	latin1_swedish_ci		No	None		      
<input type="checkbox"/>	totalcost	int(5)			No	None		      

Figure 4: Final Pre-Compiled Order Table in phpMyAdmin

The customized order table, *premadepackorder* (Figure 5), which is the most complex table in the database, comes with 25 fields. The order ID and total cost fields share the same purpose as they do in the pre-compiled order table, but the other fields play a completely different role, as follows:

- *Main Purpose* – Whether it is a personal or a business-related website.
- *Type of Business* – If it is a business website, what type of business is it?
- *Target Reader* – What types of readers will mostly read the website.

- *Number of Pages* – How many pages the customer wishes to have in the beginning. This also determines the basic cost of the website.
- *Main Colors* – The customer can input up to three main colors that he or she wishes the website to use.
- *Tone* – The tone of the website, in terms of brightness (light or dark).
- *Email contact form* – An interactive contact form which can be used to send email directly to the website owner.
- *Google Analytics* – An installation of Google’s web statistics application.
- *3 x Design Revisions* – Possibility to modify the layout of the website up to three times if the cosmetic outlook does not satisfy the customer.
- *Consulting* – Up to three hours of private consulting per week, for a period of four weeks.
- *Social Media Integration* – The website will be connected to a variety of social networking sites, such as Facebook, Twitter, Digg, StumbleUpon, so that visitors may dynamically share parts of the site content among friends.
- *Mailing List* – A mailing list attached to the website so that visitors may sign up for the list and receive updates and other messages via email.
- *Branded Promotional Banner* – A graphical banner for promoting the website on other sites (e.g. Google AdWords).
- *Instant Payment System* – Ability for visitors to make instant payments on the website through payment systems such as PayPal.
- *Shopping Cart* – Allows commercial websites to have an interactive shopping cart for organizing purchases.
- *Discussion Forum* – An installation of a discussion forum.
- *Photo Album* – A dynamic photo album for displaying photos and images on the website.
- *Google AdSense* – Allows the website owner to earn money from Google’s pay-per-click ads.
- *Other Requests* – Here the customer may submit an open request including all the additional wishes relating to the website. The developer will then assess the requests individually by responding to the customer via email, and proposing a final price for the product.
















































































































































































	Field	Type	Collation	Attributes	Null	Default	Extra	Action
<input type="checkbox"/>	custompackorderid	int(15)			No	None	auto_increment	      
<input type="checkbox"/>	mainpurpose	varchar(20)	latin1_swedish_ci		No	None		      
<input type="checkbox"/>	customerid	int(10)			No	None		      
<input type="checkbox"/>	typeofbusiness	varchar(50)	latin1_swedish_ci		No	None		      
<input type="checkbox"/>	targetreader	varchar(50)	latin1_swedish_ci		No	None		      
<input type="checkbox"/>	numberofpages	int(2)			No	None		      
<input type="checkbox"/>	pagedescriptions	text	latin1_swedish_ci		No	None		      
<input type="checkbox"/>	maincolor1	varchar(20)	latin1_swedish_ci		No	None		      
<input type="checkbox"/>	maincolor2	varchar(20)	latin1_swedish_ci		No	None		      
<input type="checkbox"/>	maincolor3	varchar(20)	latin1_swedish_ci		No	None		      
<input type="checkbox"/>	tone	varchar(10)	latin1_swedish_ci		No	None		      
<input type="checkbox"/>	contactform	varchar(3)	latin1_swedish_ci		No	None		      
<input type="checkbox"/>	googleanalytics	varchar(3)	latin1_swedish_ci		No	None		      
<input type="checkbox"/>	threerevisions	varchar(3)	latin1_swedish_ci		No	None		      
<input type="checkbox"/>	consulting	varchar(3)	latin1_swedish_ci		No	None		      
<input type="checkbox"/>	socialmedia	varchar(3)	latin1_swedish_ci		No	None		      
<input type="checkbox"/>	mailinglist	varchar(3)	latin1_swedish_ci		No	None		      
<input type="checkbox"/>	promotionalbanner	varchar(3)	latin1_swedish_ci		No	None		      
<input type="checkbox"/>	paymentsystem	varchar(3)	latin1_swedish_ci		No	None		      
<input type="checkbox"/>	shoppingcart	varchar(3)	latin1_swedish_ci		No	None		      
<input type="checkbox"/>	discussionforum	varchar(3)	latin1_swedish_ci		No	None		      
<input type="checkbox"/>	photoalbum	varchar(3)	latin1_swedish_ci		No	None		      
<input type="checkbox"/>	adsense	varchar(3)	latin1_swedish_ci		No	None		      
<input type="checkbox"/>	otherrequests	varchar(3)	latin1_swedish_ci		No	None		      
<input type="checkbox"/>	totalcost	int(5)			No	None		      

Figure 5: Final Customized Order Table in phpMyAdmin

6.2 Visual Design

The visual design consists of the structural layout of the website and the graphics that, when combined, form a cohesive design that functions as a base for the interface. Elizabeth Castro highlights the importance of designing first and developing later: “Although you can just jump in and start writing Web pages right away, it’s a good idea to first think about and design your site” (Castro 2006, 48).

Visual design, in this project, is divided into two separate parts: structure and graphics.

Structure deals with the alignment, spacing, and other arrangements that govern the shape of the website. Graphics, on the other hand, are the supporting cosmetic touches that give life to the skeletal construct.

6.2.1 Structure

The minimalist approach taken in this project led to a decision to divide the visual structure of the website into three primary elements: the header, the main body, and the footer. An outline of this structure can be seen in Figure 6 below.

Header

Main Body

Footer

Figure 6: Visual Structure

The header accommodates the company logo, the main body houses the contents of every individual page, and the footer serves as an information panel displaying basic information about the company.

The reason why this simple structure was chosen was that it eliminates distractions. Without sidebars and navigation menus, the customer can focus on the ordering process without being led to other distracting areas of the website. A large area for the main body of the content also provides space for bigger icons and font sizes. A large font size communicates clarity and focus when it is coupled with empty space.

6.2.2 Graphics

The graphical components of the visual design were first created using Adobe Illustrator and later combined in Adobe Photoshop to get a holistic visual of the final result. Working in Photoshop allows one to combine all the individual elements together and then slice them into smaller bits for web integration. The reason why all the graphics were made with Illustrator is that having the graphics as vector images allows the designer to resize and modify them without compromising image quality.

The website uses two types of images: the header logo and button images. The header logo is the top-left banner displaying the name of the company or service (twopointz3ro) along with

its slogan (Figure 7). The button images are carbon copies of a single button template drawn in Illustrator, where the text has been modified to accommodate the function of the button (Figures 8 and 9).

twopointz3ro
Websites with Zero Hassle!

Figure 7: Header Logo



Figure 8: Pre-Made Package Button

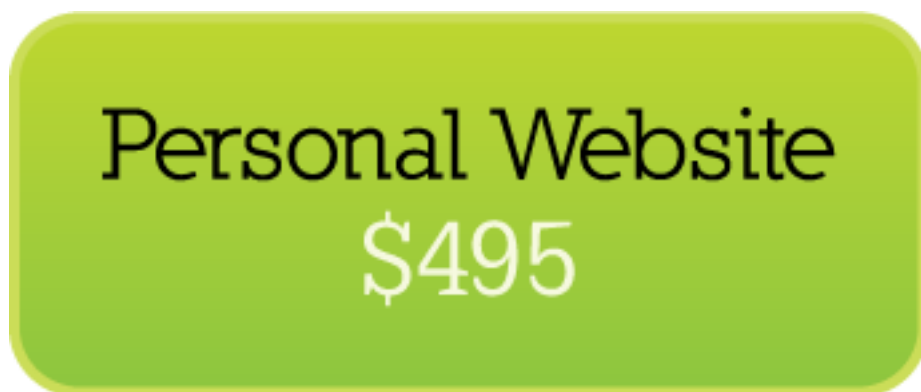


Figure 9: Personal Website Button

When all the elements are combined, the final result looks like this screen capture taken of the starting page (Figure 10):

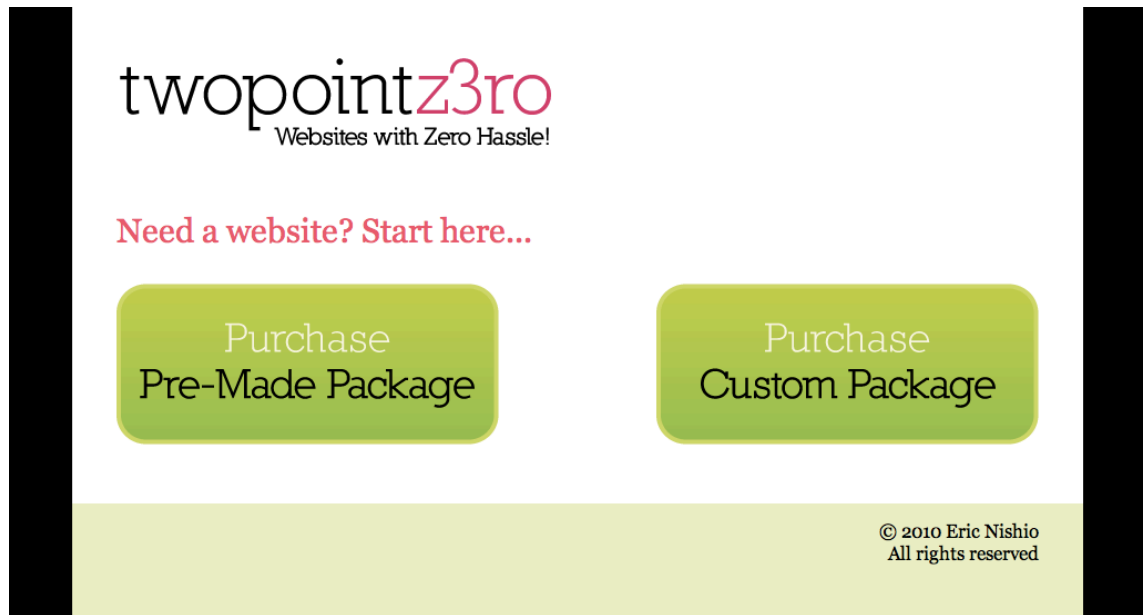


Figure 10: Basic Layout

7 Demonstration

To show how the ordering system works in practice, a test run will be conducted and screen captures will be embedded to provide a visual walkthrough. There are two demonstrations: one for the pre-compiled order and another for a customized order.

7.1 Ordering a Pre-Compiled Package

On the starting page, the customer selects *Purchase Pre-Made Package* (Figure 11).

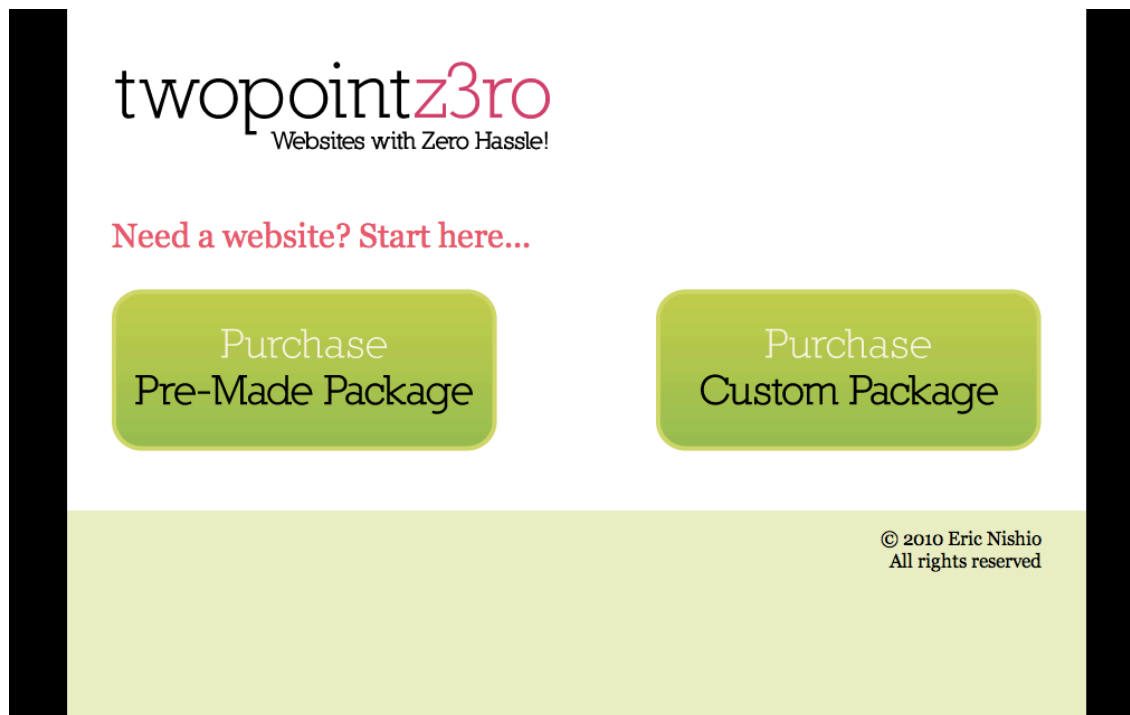


Figure 11: Pre-Compiled Order Demo, Step 1

He then chooses *Personal Website* from the list of pre-compiled websites (Figure 12).



Figure 12: Pre-Compiled Order Demo, Step 2

The next page displays information about the pre-compiled product (Figure 13). The customer then hits *Continue*.

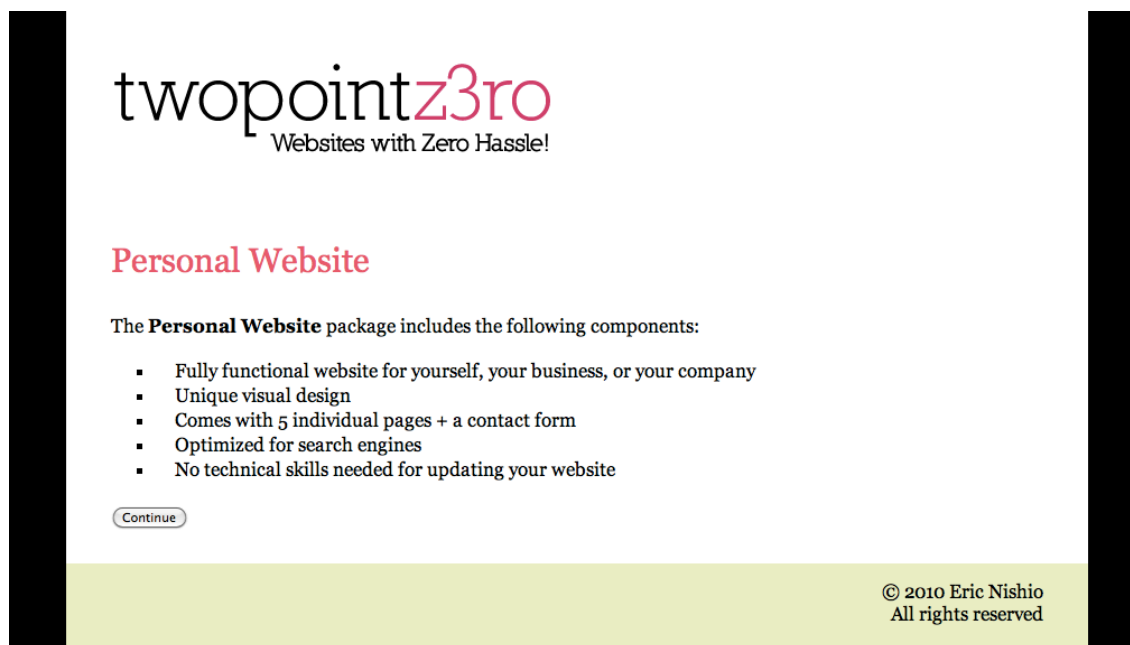



Figure 13: Pre-Compiled Order Demo, Step 3

The next screen asks the customer for his personal information, so he fills out the form, because he is using the service for the first time and does not yet have an account (Figure 14).



 Websites with Zero Hassle!

You want to order a **Personal Website**.

Total Cost: **\$495**

New Customer

Starred (*) fields are optional.

First Name

Last Name

Address 1

Address 2*

Postal Code

City

Country

Phone Number*

Gender

Email

Password

I accept the [terms and conditions](#).

Existing Customer

Email

Password

© 2010 Eric Nishio
All rights reserved

Figure 14: Pre-Compiled Order Demo, Step 4

When the form has been submitted, the system displays both the product purchased and the customer's personal information. At this point, the order is complete. There is, however, a button at the end of the page that lets the customer cancel the order (Figure 15).

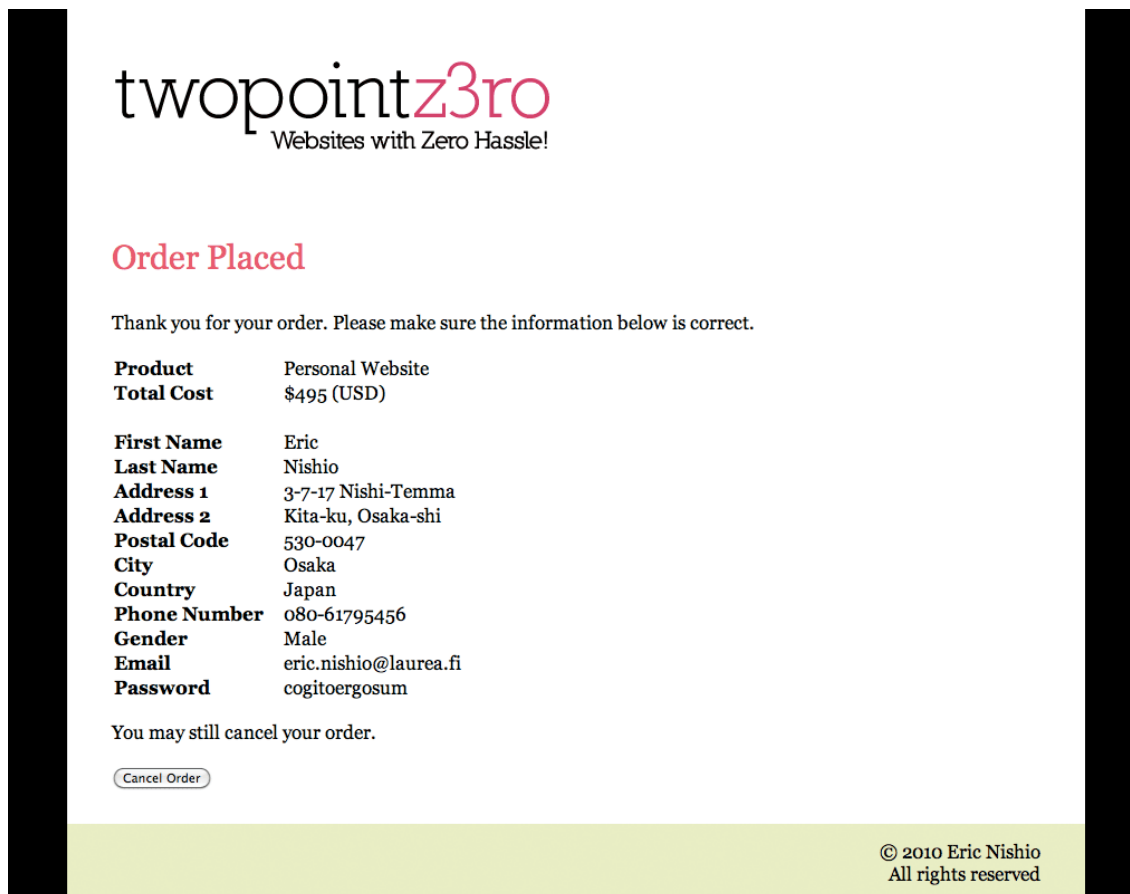


Figure 15: Pre-Compiled Order Demo, Step 5

The customer information that was stored into the database looks as follows (Figure 16).

	customerid	firstname	lastname	gender	address1	address2	postalcode	city	phonenumber	email	country	password
<input type="checkbox"/>	35	Eric	Nishio	Male	3-7-17 Nishi-Temma	Kita-ku, Osaka-shi	530	Osaka	080-61795456	eric.nishio@laurea.fi	Japan	cogitoergosum

Figure 16: Pre-Compiled Order Demo, Customer Data

And the order information is displayed as follows (17).

	premadepackorderid	customerid	productname	totalcost
<input type="checkbox"/>	12	35	Personal Website	495

Figure 17: Pre-Compiled Order Demo, Order Data

Our imaginary customer, after all, decides to cancel the order, so he clicks on the *Cancel Order* button, which takes him to a new screen that informs him that his order has been canceled (Figure 18).



Figure 18: Pre-Compiled Order Demo, Step 6

7.2 Ordering a Customized Package

This time around, the same customer decides to get a customized package, so he clicks on *Purchase Custom Package* (Figure 19).



Figure 19: Customized Order Demo, Step 1

He fills out the form by answering the questions as precisely as he can. He then clicks on *Continue* (Figure 20).

Tell us what you want...

Main Purpose Personal
 Business

Type of Business (optional)

Target Reader (optional)

Number of Pages

Describe what your pages will be about:

1. News page
2. List of products
3. Sword-fighting videos

Main Colors

Tone Light
 Dark

Extras

- Email contact form \$29
- Google Analytics (site statistics) \$29
- 3 x design revisions \$99
- Consulting (3 hours per week, for 4 weeks) \$129
- Social media integration (Facebook, Twitter, Digg, etc.) \$29
- Mailing list \$19
- Your own, branded promotional banner \$49
- Instant payment system (via PayPal or credit card) \$79
- Shopping cart \$249
- Discussion forum \$49
- Photo album \$29
- Google AdSense (get paid for ad clicks) \$39

Other Requests

Some requests will be priced separately on a case-to-case basis.

Figure 20: Customized Order Demo, Step 2

Since the customer already has an account name (email address) and password, he decides to use the login form to place the order more quickly (Figure 21).

twopointz3ro
Websites with Zero Hassle!

You want to order a **Customized Website**.

Total Cost: **\$844**
(Costs for other requests will be emailed to you separately.)

New Customer

Starred (*) fields are optional.

First Name

Last Name

Address 1

Existing Customer

Email

Password

Figure 21: Customized Order Demo, Step 3

After submitting the data to the system, the next screen displays all the information about his order as well as his personal data. The personal information retrieved from the database still matches his current information, so he does not need to do anything. The order is placed, and he will just have to wait for a response from the developers (Figure 22).

twopointz3ro

Websites with Zero Hassle!

Thank you for your order. Please make sure the information below is correct.

Your Order

Main Purpose	business
Type of Business	I craft swords and sell them.
Target Reader	reader
Number of Pages	3
Page Descriptions	1. News page 2. List of products 3. Sword-fighting videos
Main Colors	blue, silver, orange
Tone	light
Email contact form	yes
Google Analytics	
3 x design revisions	
Consulting (3 hours per week, for 4 weeks)	
Social media integration (Facebook, Twitter, Digg, etc.)	
Mailing list	yes
Your own, branded promotional banner	yes
Instant payment system	yes
Shopping cart	yes
Discussion forum	
Photo album	
Google AdSense	
Other requests	I want to upload videos by myself.
Total Cost	\$844 (USD)

Your Information

First Name	Eric
Last Name	Nishio
Address 1	3-7-17 Nishi-Temma
Address 2	Kita-ku, Osaka-shi
Postal Code	530
City	Osaka
Country	Japan
Phone Number	080-61795456
Gender	Male
Email	eric.nishio@laurea.fi
Password	cogitoergosum

You may still cancel your order.

[Cancel Order](#)

Figure 22: Customized Order Demo, Step 4

The order information is stored into the database as follows (Figure 23). (The screen capture was taken in segments, which is why the final image is uneven and distorted.)



	custompackorderid	mainpurpose	customerid	typeofbusiness	targetreader	
<input type="checkbox"/>  	22	business	35	I craft swords and sell them.	Sword buyers,	
numberofpages	pagedescriptions	maincolor1	maincolor3	tone	contactform	googleanalytics
3	1. News page 2. List of products 3. Sword-fighti...	blue	orange	light	yes	
threerevisions	consulting	socialmedia	mailinglist	promotionalbanner	paymentsystem	
			yes	yes	yes	
shoppingcart	discussionforum	photoalbum	adsense	otherrequests	totalcost	
yes				l w	844	

Figure 23: Customized Order Demo, Order Data

8 Project Phases

This project's life cycle spanned over a six-month period. Planning and preparations occupied most of the time, as it was vital to devise a coherent plan for a project that would provide real value to the ICT world.

Phase One was the first step leading to the conception of the idea for the bachelor's thesis. This period mainly consisted of attending thesis classes where senior students were instructed how to go about the final project. The teachers provided some assistance on topic selection, but the final idea for this particular project came later—after a couple of weeks of brainstorming and critical analysis.

Phase Two was when the tools were actually taken and employed. The topic was decided, and the subsequent activities centered upon refining the ideas generated around the topic (a website ordering system for the general public). At this stage, it was important to define the key concepts that would form the foundation for the project. Rather than wandering aimlessly in the woods, it was more productive to come up with a set of basic principles to work with. This is where the notions of minimalism and user-friendliness entered the scene.

Phase Three came rather fast, and although the spring was yet to be on the horizon (the treetops were still nicely covered with powder snow) the development entered a productive mode. More concepts were adopted, and the first UML models were created. They were, in fact, so elementary in nature that it would baffle the reader if the originals were attached to this report. They served as a preliminary basis for the project. The advantage of taking action

as early as possible is the fact that one is able to produce something tangible and concrete. Regardless of how bad the result is, it is better to have something rather than to be empty-handed, because it allows the developer to take a critical look at the project and see what needs to be changed and and developed further.

There were some inactive periods between Phase Three and Phase Four, primarily due to an unyielding spring semester when the schedule was filled to the rim. Although the initial plan was to finish the project by the end of the spring semester in May, certain time-related complications hindered the meeting of the intended deadline. Hence the deadline was rescheduled for September. Development resumed after the spring finals.

In Phase Four, the UML models were refined, and it was finally time to start developing the system itself. First of all, the visual layout of the website. This was done using Adobe Illustrator to create the graphics that were then assembled together in Photoshop to create a visual mock-up of the final design. When the graphics were satisfactory, it was time to construct the framework for the website in HTML and CSS. This framework was to function as the foundational structure around which the ordering system itself would be built.

Now that the base was ready, Phase Five would finally introduce the implementation of the MySQL database and the PHP scripts that would allow users to place orders on the site. The MySQL database was first designed according to the UML blueprints that were made earlier so that it would accommodate all the PHP scripts required for successfully operating the system. Finally, the PHP scripts were written and embedded into the HTML code that was written earlier in Phase Four. This was the final phase of the actual development cycle of this project. After critically testing the system, it was time to record the results and focus on the report.

9 Further Development

The work that has been done so far only marks the beginning of this project. Although the system itself is fully functional and ready to be implemented for commercial use, it is only the first iteration of the system. Using the system in a real, commercial context will give rise to a multitude of new concepts and ideas that need to be taken into account.

The first and foremost aspect that requires attention is how well customized products are able to do. It is certainly impossible to satisfy all types of customers, but it is impossible to successfully predict what reality will bring. The customization tool employed on the website may not be fully optimized, so this has to be established through trial and error, by observing how customers react to the tool, and what *other requirements* they decide to add to the list.

An internal test run will never suffice as a realistic measure for evaluating the applicability and success of the system. It is, therefore, imperative that the system be tested in practice and with real customers.

Another aspect that was impossible to implement in the first iteration of the project was embedded payment methods. Since twopointz3ro is still merely a work-in-progress, it is not eligible for instant payment services provided by companies such as PayPal. While it is possible to run the business without the assistance of such instant payment services by handling the payments manually via email, the lack of these payment services can be a major obstacle (and turn-off) for a great number of customers. However, in order to employ these payment services, the system has to be fully implemented and made available for commercial use. Only then will twopointz3ro be able to reach its full potential and credibility as a trustworthy service provider.

10 Conclusion

The project was a success, and it met the requirements that were established during the planning phase. The ordering system was built around the concepts of minimalism, simplicity, and user-friendliness. From the developer's standpoint, all the criteria have been satisfied. There is still much to do in terms of releasing the service for the public, but as a mechanical tool it serves its purpose very well.

The greatest thing about developing this system was the liberty of creating a commercially functional system from scratch. Creating it individually further amplified the sense of achievement in that every single step was taken and successfully completed by a single person. Having completed this task alone instills in the student a tremendous amount of self-confidence, because he finally realizes what he is capable of accomplishing. I believe that the bachelor's thesis is one of the critical and necessary steps that allow a student to mature into a full-fledged professional, ready to confront the world of opportunity.

References

Castro, E. 2006. HTML, XHTML & CSS. California: Peachpit Press.

Hunt, B. 2008. Save the Pixel: The Art of Simple Web Design.

Krug, S. 2005. Don't Make Me Think: A Common Sense Approach to Web Usability. California: New Riders.

Miles, R. & Hamilton, K. 2006. Learning UML 2.0. California: O'Reilly Media, Inc.

Rowse, D. 2010. How to Be a More Relational Blogger. Retrieved on July 26, 2010.
<http://www.probblogger.net/archives/2010/01/26/how-to-be-a-more-relational-blogger-tips-for-new-and-established-bloggers/>

Snell, S. 2010. 30 Examples Of Well Designed Ecommerce Websites. Retrieved on June 24, 2010. <http://spyrestudios.com/30-examples-of-well-designed-ecommerce-websites/>

Solomon, G. & Schrum, L. Web 2.0: New tools, New Schools. Washington DC: International Society for Technology in Education (ISTE).

Welling, L. & Thomson, L. 2009. PHP and MySQL Web Development. United States: Pearson Education, Inc.

Figures

Figure 1: Use-Case Diagram	17
Figure 2: Original Database Diagram.....	19
Figure 3: Final Customer Table in phpMyAdmin.....	20
Figure 4: Final Pre-Compiled Order Table in phpMyAdmin	20
Figure 5: Final Customized Order Table in phpMyAdmin	22
Figure 6: Visual Structure	23
Figure 7: Header Logo.....	24
Figure 8: Pre-Made Package Button.....	24
Figure 9: Personal Website Button	24
Figure 10: Basic Layout	25
Figure 11: Pre-Compiled Order Demo, Step 1	26
Figure 12: Pre-Compiled Order Demo, Step 2	26
Figure 13: Pre-Compiled Order Demo, Step 3	27
Figure 14: Pre-Compiled Order Demo, Step 4	28
Figure 15: Pre-Compiled Order Demo, Step 5	29
Figure 16: Pre-Compiled Order Demo, Customer Data.....	29
Figure 17: Pre-Compiled Order Demo, Order Data	29
Figure 18: Pre-Compiled Order Demo, Step 6	30
Figure 19: Customized Order Demo, Step 1	30
Figure 20: Customized Order Demo, Step 2	31
Figure 21: Customized Order Demo, Step 3	32
Figure 22: Customized Order Demo, Step 4	33
Figure 23: Customized Order Demo, Order Data.....	34

Attachment 1 Benchmarking

To gain better insight on the usability and effectiveness of a web-based ordering system, an analysis will be conducted. The benchmark will analyze a number of different e-commerce websites that specialize in selling products to customers, and evaluate their strengths and weaknesses. The underlying purpose for gathering this data is to make the final product as good as possible.

The aim of the project is to develop the best interface and functionality in the market for a web design provider, so the analysis will focus on collecting useful data only from other successful and highly acclaimed websites and companies. Some of the websites included in this benchmark have been rated as examples of well-designed e-commerce websites by web design company spyrestudios (30 Examples Of Well Designed Ecommerce Websites 2010).

The analysis will also not be limited to websites of other web service providers, and analyses will be conducted on a variety of website types ranging from restaurant websites to on-line clothing stores. A diverse test group enables us to draw inspiration from unrelated e-commerce websites, providing us with new ideas leading to innovative solutions.

Case 1: Piccadilly

Description: Website of a family restaurant
URL: www.piccadilly.com

Strengths:

- clean web design
- easy to navigate
- current discounts displayed visibly with detailed images on the front page
- core functions (menu and location finder) are easily accessible
- location finder

Weaknesses:

- food menu includes only partial images of foods

Case 2: Incase

Description: Seller of cases for Apple products
URL: www.goincase.com

Strengths:

- minimalist web design (loyal to the Apple brand)
- product categories are easy to find
- products are categorized logically (intuitive product browser)

Weaknesses:

- front page does not display the company's most popular products (poor product marketing)
- heavy Flash animations on the front page may cause slower computers to slow down
- main navigation bar (at the top of the page) has ambiguous link descriptions

Case 3: Storenvy

Description: A company primarily offering free online stores
URL: www.storenvy.com

Strengths:

- trendy web design
- a pleasure to look at the large typography

- offers something for free

Weaknesses:

- first time visitors have a hard time discerning what products or services the company offers
- lack of focus; the company's products and services are too diverse to cater to a distinct niche market
- site navigation has not been optimized; there are too many places to click scattered over any given page

Case 4: Lulu

Description: Lets individuals publish their own e-books and earn money from them

URL: www.lulu.com

Strengths:

- clean and easy-to-read web design
- unique service (well-defined niche)
- site navigation is very intuitive

Weaknesses:

- increasing number of side products unrelated to publishing e-books

Case 5: coast

Description: Online store selling women's evening wear

URL: www.coast-stores.com

Strengths:

- clean web design
- special navigation options on the front page (eg. "Find the perfect dress")
- navigation bar (at the top of the page) is divided into logical categories
- versatile product browser (many criteria to choose from)
- detailed product images (high quality zoom, multiple angle shots)

Weaknesses:

- slower computers might have problems browsing the site because of Flash effects and functions
- adding items to the shopping bag is tricky because there's another option called "Add items to fitting room" which is a different function altogether

Case 6: TemplateMonster.com

Strengths:

- large selection of templates to choose from
- template screenshots are displayed explicitly
- templates have their individual pages
- screenshots are big and detailed
- special templates for content management systems and blog platforms, such as Joomla and WordPress
- regular and buyout purchase options (buyout will remove the template from the selection)
- possibility to customize the template

Weaknesses:

- pages are bloated with text
- lack of focus
- the site does not tell the customer in what ways templates can be customized

Case 7: HooverWebDesign.com

Strengths:

- defined niche: business and personal websites

Weaknesses:

- poor design on the main website; lacks trustworthiness
- AdSense advertisement are displayed on every page; this directs visitors to other (com-peting) websites
- excess text; irrelevant content distracts visitors
- shopping cart is unclear

Case 8: TotalCreation.co.uk

Strengths:

- options are given at the top of the front page
- example prices given on the front page

Weaknesses:

- anchor texts not intuitive enough
- poor color scheme
- design not professional-looking
- excess amount of text

Case 9: CoolCreation.co.uk

Strengths:

- sleek and modern design
- compelling sample websites

Weaknesses:

- hard to get on the right path at first
- too many vague options given
- employs IT abbreviations

Case 10: The WebsiteGurus.com

Strengths:

- professional-looking website
- products are easy to find
- promotes a particular service (“Select a design, send us your content, we do the rest!”)
- large variety of sample layouts

Weaknesses:

- pricing is unclear

Case 11: Chris-Smith-Web.com

Strengths:

- easy website navigation

Weaknesses:

- excess amount of text
- tone too personal
- design too colorful

Case 12: Tradepie.com

Strengths:

- options displayed on the front page

Weaknesses:

- design outdated
- too many icons and strings of text on a single page
- ambiguous options
- thronged with IT terminology and abbreviations

Case 13: AffordableWebdesign.com**Strengths:**

- good domain name
- clear niche/focus

Weaknesses:

- cheap-looking design (and even too corporate in tone)
- no clear path for purchasing a product
- too much text
- small font

Case 14: Design2Kill.com**Strengths:**

- testimonials

Weaknesses:

- cheap-looking design
- too much text on a single page
- Google Ads nestled between page elements (poor branding)
- small font
- most pages do not offer any coherent information

Case 15: Lineofsite.co.uk**Strengths:**

- pricing is shown upfront
- clear navigation menu

Weaknesses:

- unprofessional-looking site layout
- poor sample designs

Case 16: TrulyUniqueWeb.com**Strengths:**

- offers and prices given on the front page

Weaknesses:

- design looks extremely unprofessional
- company seems unreliable

Case 17: Webdesignerexpress.com**Strengths:**

- simple website layout
- focuses on cheap prices (good on-site promotions)

Weaknesses:

- text hard to read sometimes

Case 18: KarmickSolutions.com**Strengths:**

- clean graphical template

Weaknesses:

- no focus on web design (perplexes the customer and jeopardizes reliability)
- difficult to buy a website (unclear path)

Case 19: Easy3StepSites.com**Strengths:**

- clear path for purchasing a website solution
- focused on ease and clarity
- adequate amount of text

Weaknesses:

- site layout does not look overly professional

Case 20: Woodeewebdesign.com**Strengths:**

- clear product bundles with prices displayed on the front page
- easy navigability
- written in general English (minimal IT knowledge required)

Weaknesses:

- dark color scheme

Attachment 2 Interview

Interviewee:

Virpi Kaartti, Lecturer and Project Manager at Laurea University of Applied Sciences, Leppävaara (former customer of a web development company)

(Translated into English from Finnish)

1. What are your favorite websites in terms of user interfaces and usability?
 - Facebook.com, LinkedIn.com
 - websites that have a distinct purpose and focus
2. What is your favorite online store?
 - Amazon (although it is partly unclear and cluttered)
3. What information should be displayed on the front page of an online store?
 - a slogan or something special related to the brand
 - references
 - testimonials
5. How many clicks are you ready to perform in order to buy a product or a service?
 - it does not matter as long as the buying process is clear
 - every step should be clearly indicated (possible progress bars)
 - lengthy forms should be avoided
 - needed: the ability to save your progress and return to it later
 - needed: help functions and a notification that you can expect an answer to your question within X hours
6. Does it suffice to have PayPal as the only payment method?
 - difficult to say
 - it may limit the business substantially
 - what payment policies do they have? (installments)
7. Which factors make an online store reliable/trustworthy?
 - things related to paying
 - who the owner is (company information)
 - certificates that can be identified by the client (or descriptions of the certificates)
8. What do you want the service package to include by default?
 - project analysis (what the project is about)
 - implementation/installation
 - testing
 - troubleshooting
9. What were some of your good experiences during the development process of the Laurea SID website?
 - all parties were reliable
 - the costs were poorly estimated in the first implementation
10. What are some aspects with the Laurea SID project that could have been better?
 - written documentation in the first implementation
 - analysis
 - cost estimates
 - schedule
 - personality
 - register in communication (understandable for all parties)
11. What additional services should a web development company offer?

- maintenance
- technical management (so that the client would not have to do it alone)
- helpdesk
- contingency opportunities

Supplementary Data

- every client should have his own path
- personal websites should contain a photo album, a synchronized calendar, privacy rules (eg. visible only for family and friends), feeds from other sites
- basic modules should be ready
- possibility to choose from a basic package and a customized package
- physical communication is not necessary; but it would be optimal to have the first session in person
- the client should be “forced” to think about requirements
- concept creation should be done separately
- the client will have to think within the context of the ordering system

Attachment 3 HTML and CSS

In order to bring all the elements created in Illustrator and Photoshop together, the individual images must be laid out using HTML to first design the website structure and then place the graphics appropriately.

Since the website involves many copies of the same website template, the template itself will be created by combining different elements together using PHP. This is because HTML alone is not capable of combining bits of HTML together in a cohesive way. The first step, therefore, is to create the elementary page elements in HTML and PHP. Every bit of code in this project is written with a simple Mac text editor called TextEdit.

Elements

header.php

The header.php template starts by defining the basic items that must be present on every page. With the *!DOCTYPE* tag it defines which HTML rules are to be used when interpreting the page in an Internet browser.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<html>
<head>
```

The *title* field includes a string of PHP code to allow individual pages to have their own, unique titles instead of being limited to a collective title, such as the company name.

```
<title><?php echo $page_title; ?></title>
```

The *meta name* tags here define what type of website it is. Search engines and other applications that analyze websites can then easily extract this information. These small bits of code are important in optimizing the website for search engines, because search engines are robots that are designed to devour website information according to pre-determined algorithms.

```
<meta name="description" content="Create your own website." />
<meta name="keywords" content="website, purchase, web developer, web
2.0" />
<meta name="author" content="Eric Nishio" />
```

Meta http-equiv defines which character set to use. To avoid garbling, this project uses UTF-8, which supports almost every type of alphabet and script used on the Internet.

```
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
```

The *link rel* tag defines which CSS style sheet will be used throughout the website.

```
<link rel="stylesheet" type="text/css" href="style.css" />
</head>
```

```
<body>
```

Now the header template must be divided into the appropriate CSS elements used by every page. The *wrapper* class is the section of the website where all the elements will be collected; it is basically the area that houses all the text, images, and other visual material on a web page. It is used in order to differentiate the main part from the black edges of the

page (that do not serve any practical purpose content-wise). The *header* class begins at the top-most area of the page where the website logo (*logo-area*) will be placed.

```
<div id="wrapper"> <!-- Wrapper Starts -->

<div id="header"> <!-- Header Starts -->

<div id="logo-area">
<a href="index.php"></a>
</div>

</div> <!-- Header Ends -->
```

footer.php

The footer template starts with a declaration that starts the *footer* class.

```
<div id="footer"> <!-- Footer Starts -->

<div style="float: right;">
```

The following PHP code inserts the current year to be used in the copyright declaration.

```
&copy; <?php echo date("Y"); ?> Eric Nishio<br />
All rights reserved
</div>
```

The footer and wrapper classes must be properly ended with a `</div>` so that the page would not be unfinished or lacking any code.

```
</div> <!-- Footer Ends -->

</div> <!-- Wrapper Ends -->

</body>
</html>
```

Page Template

A page template is used for creating new pages. It includes all the necessary pieces of code to maintain a cohesive structure, so that no individual pages will have different style sets or other deviations that would render them stylistically incongruent.

```
<?php
```

The `session_start()` function is used on every page to initiate a session that will be used when passing variables from one page to another.

```
session_start();
```

The title of the individual page is set by defining the `$page_title` variable, which will be then used by the `header.php` element defined earlier.

```
$page_title = "twopointz3ro - Websites with Zero Hassle!";
?>
```

The following code inserts the `header.php` element onto the page, so that it will be visible when viewing the page with an Internet browser.

```
<?php include("header.php"); ?>
```

The *main* class is where the main content (headings, text, images) of the website is placed.

```
<div id="main">

<h1>Main heading of the page</h1>

<p>
Main content comes here.
</p>

</div>
```

Last, the *footer.php* element is placed at the end of a page.

```
<?php include("footer.php"); ?>
```

CSS Style Sheet

The CSS style sheet is a file containing all the cosmetic settings for the entire website. It defines how all the classes are aligned and displayed on a page, what fonts to use and in what sizes, and what colors to employ. It is an effective way to keep the layout of a website organized and congruent throughout.

First, the *html* and *body* tags are defined by setting the background color to black (#000), the text color to black (since the wrapper class is entirely white), font size to 17 pixels, and the font family to Georgia, or if not available, Times New Roman, Times, or serif.

```
html, body {
    background-color: #000;
    color: #000;
    margin: 0;
    padding: 0;
    font-family: Georgia, "Times New Roman", Times, serif;
    font-size: 17px;
}
```

The margins for headings are set to no margins at all (0).

```
h1, h2, h3, h4, h5, h6 {
    margin: 0;
    padding: 0;
}
```

The margins and all borders for images are set to zero.

```
img {
    margin: 0;
    padding: 0;
    border: 0;
}
```

The two headings are stylized to suit the basic font size.

```
h1 {
    color: #f6416b;
    font-size: 30px;
}
```



```

        font-weight: normal;
        margin-bottom: 1em;
    }

h2 {
    color: #000;
    font-size: 25px;
    font-weight: normal;
    margin-top: 1.5em;
    margin-bottom: 1em;
}

```

All paragraphs are set to have a line spacing of 1.4 lines.

```

p {
    line-height: 1.4em;
}

```

Link colors are set to #f6416b when normal and #f3849e when moused over, and decorations are set to none.

```

a {
    color: #f6416b;
    text-decoration: none;
}

a:hover {
    color: #f3849e;
}

```

All tables are set to not have any borders or margins, and the cells should always start at the top, not the middle.

```

table, tr, td {
    border: 0;
    margin: 0;
    padding: 0;
    vertical-align: top;
}

```

The table cells are sized as follows.

```

td.first {
    width: 150px;
}

td.custom {
    width: 250px;
}

```

Two green color settings defined for displaying the total cost of a product and for miscellaneous bits of text.

```

.green {
    color: #090;
}

.totalcost {
    color: #090;
    font-size: 35px;
}

```

```
}

```

The following are the settings for the wrapper class.

```
#wrapper {
    width: 912px;
    margin: auto;
    padding: 0;
    background-color: #fff;
}

#header {
}

#logo-area {
    margin: 0;
}
```

The size and margin settings for the main content class are defined as follows.

```
#main {
    width: 832px;
    padding: 30px 40px 30px 40px;
    float: left;
}

#main ul {
    list-style-type: square;
    margin-top: 0;
    margin-bottom: 0;
}

#main ol {
    margin-top: 0;
    margin-bottom: 0;
}

#main li {
    margin-bottom: 1em;
    line-height: 1.3em;
    padding-left: 1em;
    margin: 0;
}
```

The input, password and textarea fields in a form field are stylized to adapt better to the large fonts defined earlier.

```
input[type="text"], input[type="password"] {
    width: 180px;
    border: solid 1px #e6dcdc;
    padding: 5px;
    font-family: Arial, Helvetica, sans-serif;
    font-size: 13px;
}

textarea {
    width: 400px;
    height: 100px;
    border: solid 1px #e6dcdc;
    padding: 5px;
}
```

```
        font-family: Arial, Helvetica, sans-serif;
        font-size: 13px;
    }

    label {
        display: block;
        float: 1em;
    }
```

The footer class is defined to look appropriate against the main body and the header element.

```
#footer {
    color: #000;
    height: 200px;
    width: 832px;
    padding: 15px 40px 15px 40px;
    background-color: #e6f1c0;
    clear: both;
    text-align: right;
}

#footer a:hover {
    color: #cf9393;
}
```

Attachment 4 MySQL Database

MySQL Database

The MySQL database is used for storing customer information as well as data related to the purchases made by customers. Every individual order will be stored into the database for as long as is required by the developer and customer. The PHP scripts (detailed in the next section) will function directly in conjunction with the MySQL database, which makes it important to create it accurately for accommodating the data submitted from the website.

There are a total of three tables: the customer table, pre-compiled order table, and customized order table.

Customer Table

The following SQL command creates the customer table with 12 fields, and adds *customerid* as its primary key with an auto increment function. The default character set is UTF-8, to blend in with the HTML code.

```
CREATE TABLE `customer` (
  `customerid` int(10) NOT NULL AUTO_INCREMENT,
  `firstname` varchar(15) NOT NULL,
  `lastname` varchar(30) NOT NULL,
  `gender` varchar(6) NOT NULL,
  `address1` varchar(30) NOT NULL,
  `address2` varchar(30) NOT NULL,
  `postalcode` varchar(15) NOT NULL,
  `city` varchar(25) NOT NULL,
  `phonenumber` varchar(40) NOT NULL,
  `email` varchar(50) NOT NULL,
  `country` varchar(20) NOT NULL,
  `password` varchar(20) NOT NULL,
  PRIMARY KEY (`customerid`)
) ENGINE=MyISAM DEFAULT CHARSET=utf-8 AUTO_INCREMENT=35;
```

Pre-Compiled Order Table

The following SQL code creates the pre-compiled order table, including four fields, and sets *premadepackorderid* as its primary key with an auto increment function. The default character set is UTF-8, to blend in with the HTML code.

```
CREATE TABLE `premadepackorder` (
  `premadepackorderid` int(15) NOT NULL AUTO_INCREMENT,
  `customerid` int(10) NOT NULL,
  `productname` varchar(50) NOT NULL,
  `totalcost` int(5) NOT NULL,
  PRIMARY KEY (`premadepackorderid`)
) ENGINE=MyISAM DEFAULT CHARSET=utf-8 AUTO_INCREMENT=12;
```

Customized Order Table

The following SQL code creates the customized order table, including 25 fields, and sets *custompackorderid* as its primary key with an auto increment function. The default character set is UTF-8, to blend in with the HTML code.

```
CREATE TABLE `custompackorder` (
  `custompackorderid` int(15) NOT NULL AUTO_INCREMENT,
  `mainpurpose` varchar(20) NOT NULL,
```

```
`customerid` int(10) NOT NULL,  
`typeofbusiness` varchar(50) NOT NULL,  
`targetreader` varchar(50) NOT NULL,  
`numberofpages` int(2) NOT NULL,  
`pagedescriptions` text NOT NULL,  
`maincolor1` varchar(20) NOT NULL,  
`maincolor2` varchar(20) NOT NULL,  
`maincolor3` varchar(20) NOT NULL,  
`tone` varchar(10) NOT NULL,  
`contactform` varchar(3) NOT NULL,  
`googleanalytics` varchar(3) NOT NULL,  
`threerevisions` varchar(3) NOT NULL,  
`consulting` varchar(3) NOT NULL,  
`socialmedia` varchar(3) NOT NULL,  
`mailinglist` varchar(3) NOT NULL,  
`promotionalbanner` varchar(3) NOT NULL,  
`paymentsystem` varchar(3) NOT NULL,  
`shoppingcart` varchar(3) NOT NULL,  
`discussionforum` varchar(3) NOT NULL,  
`photoalbum` varchar(3) NOT NULL,  
`adsense` varchar(3) NOT NULL,  
`otherrequests` varchar(3) NOT NULL,  
`totalcost` int(5) NOT NULL,  
PRIMARY KEY (`custompackorderid`)  
) ENGINE=MyISAM DEFAULT CHARSET=utf-8 AUTO_INCREMENT=22;
```

Attachment 5 PHP Scripts

PHP Scripts

The PHP scripts in this project control the ordering process. The HTML, CSS, and MySQL parts are merely the foundation that allows the website to store information in the database. HTML and CSS, together with the graphics, form the visual interface that will be used by customers. The PHP scripts, however, are the “brains” of the system, controlling how the system responds to the customers’ actions.

Pre-Compiled Package

This hypothetical order shows the PHP scripts that are used when ordering a pre-compiled personal website package.

The order is started with a `session_start()` declaration, which allows the system to store variables temporarily. Next, the system stores the product information the customer wishes to order.

```
session_start();
$_SESSION["order_what"] = "Personal Website";
$_SESSION["totalcost"] = 349;
```

This information is then passed onto the next page (`pre-made-order.php`) by the following HTML form function.

```
<form action="pre-made-order.php" method="post">
<input type="submit" value="Continue" />
</form>
```

Then the system establishes a connection to the database, checks that it works, and redefines the previously acquired variables from the `$_SESSION` array.

```
// Connect to database
$user = "root";
$password = "";
$host = "localhost";
$dbname = "twopointz3ro";
$link = mysql_connect($host, $user, $password);

// Validate database connection
if (!$link) {
    die("No connection: " . mysql_error());
}

// Connect to database
mysql_select_db($dbname, $link);

// Previous variables
$order_what = $_SESSION["order_what"];
$totalcost = $_SESSION["totalcost"];
```

Next, the customer is prompted to fill out the form for submitting his or her personal information. Existing customers also have the option to enter their email address and password for instant ordering, but that will be covered in the next section when our hypothetical customer purchases a customized package.

```
echo "<form action=\"pre-made-register.php\" method=\"post\">";
```

```

echo "<table>";
echo "<tr>";
echo "<td class=\"first\"><strong>First Name</strong></td>";
echo "<td><input type=\"text\" name=\"firstname\" ></td>";
echo "</tr>";
echo "<tr>";
echo "<td class=\"first\"><strong>Last Name</strong></td>";
echo "<td><input type=\"text\" name=\"lastname\" ></td>";
echo "</tr>";
echo "<tr>";
echo "<td class=\"first\"><strong>Address 1</strong></td>";
echo "<td><input type=\"text\" name=\"address1\" ></td>";
echo "</tr>";
echo "<tr>";
echo "<td class=\"first\"><strong>Address 2*</strong></td>";
echo "<td><input type=\"text\" name=\"address2\" ></td>";
echo "</tr>";
echo "<tr>";
echo "<td class=\"first\"><strong>Postal Code</strong></td>";
echo "<td><input type=\"text\" name=\"postalcode\" ></td>";
echo "</tr>";
echo "<tr>";
echo "<td class=\"first\"><strong>City</strong></td>";
echo "<td><input type=\"text\" name=\"city\" ></td>";
echo "</tr>";
echo "<tr>";
echo "<td class=\"first\"><strong>Country</strong></td>";
echo "<td><input type=\"text\" name=\"country\" ></td>";
echo "</tr>";
echo "<tr>";
echo "<td>&nbsp;</td>";
echo "<td>&nbsp;</td>";
echo "</tr>";
echo "<tr>";
echo "<td class=\"first\"><strong>Phone Number</strong>*</td>";
echo "<td><input type=\"text\" name=\"phonenumber\" ></td>";
echo "</tr>";
echo "<tr>";
echo "<td class=\"first\"><strong>Gender</strong></td>";
echo "<td><select name=\"gender\"><option
value=\"Male\">Male</option><option
value=\"Female\">Female</option></select></td>";
echo "</tr>";
echo "<tr>";
echo "<td>&nbsp;</td>";
echo "<td>&nbsp;</td>";
echo "</tr>";
echo "<tr>";
echo "<td class=\"first\"><strong>Email</strong></td>";
echo "<td><input type=\"text\" name=\"email\" ></td>";
echo "</tr>";
echo "<tr>";
echo "<td class=\"first\"></td>";
echo "<td><input type=\"text\" name=\"email_confirm\" ></td>";
echo "</tr>";
echo "<tr>";
echo "<td>&nbsp;</td>";
echo "<td>&nbsp;</td>";
echo "</tr>";
echo "<tr>";

```

```

echo "<td class=\"first\"><strong>Password</strong></td>";
echo "<td><input type=\"password\" name=\"password\" ></td>";
echo "</tr>";
echo "<tr>";
echo "<td class=\"first\"></td>";
echo "<td><input type=\"password\" name=\"password_confirm\" ></td>";
echo "</tr>";
echo "<tr>";
echo "<td>&nbsp;</td>";
echo "<td>&nbsp;</td>";
echo "</tr>";
echo "<tr>";
echo "<td class=\"first\"><strong></strong></td>";
echo "<td><input type=\"checkbox\" name=\"acceptconditions\"
value=\"accept\" > I accept the <a
href=\"termsandconditions.php\">terms and conditions</a>.</td>";
echo "</tr>";
echo "<tr>";
echo "<td>&nbsp;</td>";
echo "<td>&nbsp;</td>";
echo "</tr>";
echo "<tr>";
echo "<td></td>";
echo "<td><input type=\"submit\" value=\"Confirm Order\" /></td>";
echo "</tr>";
echo "</table>";
echo "</form>";

```

The personal information then gets passed onto the registration page (pre-made-register.php).

The system retrieves the form data by accessing the `$_POST` array.

```

$firstname = $_POST["firstname"];
$lastname = $_POST["lastname"];
$gender = $_POST["gender"];
$address1 = $_POST["address1"];
$address2 = $_POST["address2"];
$postalcode = $_POST["postalcode"];
$city = $_POST["city"];
$phonenumber = $_POST["phonenumber"];
$email = $_POST["email"];
$email_confirm = $_POST["email_confirm"];
$country = $_POST["country"];
$password = $_POST["password"];
$password_confirm = $_POST["password_confirm"];
$acceptconditions = $_POST["acceptconditions"];

```

Next, the system checks if the information is acceptable. There are different scripts for checking the validity of the email address, checking that the two emails and passwords match each other, confirming that every compulsory field has been filled, and checking that the terms and conditions have been accepted.

```

// Duplicate email check
$query_checkdupeemail = "SELECT * FROM customer WHERE email =
'$email'";
$get_checkdupeemail = mysql_query($query_checkdupeemail, $link) or
die("Error in query Check Dupe Email.");
$checkdupeemail = mysql_num_rows($get_checkdupeemail);
if ($checkdupeemail >= 1) {

```



```

        echo "<p>Email already in use.</p><form><input
type=\"button\" value=\"Go Back\" onClick=\"history.go(-1);return
true;\" /></form>";
    }

    // Email validation
    if (filter_var($email, FILTER_VALIDATE_EMAIL)) {
        $email_validi = TRUE;
    } else {
        $email_validi = FALSE;
    }

    if ($email_validi == FALSE) {
        echo "<p>Incorrect email.</p><form><input type=\"button\"
value=\"Go Back\" onClick=\"history.go(-1);return true;\" /></form>";
    }

    // Confirm that all fields have been filled
    if (($firstname == "") || ($lastname == "") || ($gender == "") ||
($address1 == "") || ($postalcode == "") || ($city == "") || ($email
== "") || ($country == "") || ($password == "")) {
        echo "<p>Please fill out all mandatory
fields.</p><form><input type=\"button\" value=\"Go Back\"
onClick=\"history.go(-1);return true;\" /></form>";
    }

    // Confirm email
    if (($email != $email_confirm) && ($email != "")) {
        echo "<p>Email addresses don't match.</p><form><input
type=\"button\" value=\"Go Back\" onClick=\"history.go(-1);return
true;\" /></form>";
    }

    // Confirm password
    if (($password != $password_confirm)) {
        echo "<p>Passwords do not match.</p><form><input
type=\"button\" value=\"Go Back\" onClick=\"history.go(-1);return
true;\" /></form>";
    }

    // Confirm acceptance of terms and conditions
    if ($acceptconditions != "accept") {
        echo "<p>You must accept the terms and conditions before
you can place any orders.</p><form><input type=\"button\" value=\"Go
Back\" onClick=\"history.go(-1);return true;\" /></form>";
    }

```

If the submitted data gets through, the system first stores the customer information into the database (*customer* table), and creates a variable for the customer ID so that the order can be stored too.

```

// Store customer into database
$insert_customer = "INSERT INTO customer (firstname, lastname, gender,
address1, address2, postalcode, city, phonenumber, email, country,
password)
VALUES ('$firstname', '$lastname', '$gender', '$address1',
'$address2', '$postalcode', '$city', '$phonenumber', '$email',
'$country', '$password')";
mysql_query($insert_customer, $link) or die("Error storing
customer.");

```

```

$query_newcustomerid = "SELECT * FROM customer WHERE email =
'$email'";
$get_newcustomerid = mysql_query($query_newcustomerid, $link);
$newcustomerid = mysql_result($get_newcustomerid, 0, "customerid");

```

Finally, the system stores the order into the database (*premadepackorder* table) while retrieving the product name and total cost from the `$_SESSION` array.

```

// Previously set data
$order_what = $_SESSION["order_what"];
$totalcost = $_SESSION["totalcost"];

// Store order into database
$insert_order = "INSERT INTO premadepackorder (customerid,
productname, totalcost)
VALUES ('$newcustomerid', '$order_what', '$totalcost')";
mysql_query($insert_order, $link) or die("Error storing order.");
$_SESSION["neworderid"] = mysql_insert_id();

```

When the order has been stored into the database, the stored information is shown on the page for the customer to see if there are any mistakes. At this point the order has been successfully stored into the database, meaning that the customer does not need to confirm it anymore. In case the customer wants to cancel the order, he or she can still use the *Cancel Order* button to delete the order.

For canceling the order, the system uses the following lines of code. First, it retrieves the order ID from the `$_SESSION` array, and deletes the information stored under that order ID.

```

// Previous variables
$neworderid = $_SESSION["neworderid"];

// Cancel order
$cancel_order = "DELETE FROM premadepackorder WHERE premadepackorderid
= '$neworderid'";
mysql_query($cancel_order, $link) or die("Error canceling order.");

```

Customized Package

This hypothetical order shows the PHP scripts that are used when ordering a pre-compiled personal website package.

As with pre-compiled orders, customized orders also start with a `session_start()` declaration, which allows the system to store variables for as long as the person remains perusing the website. Next, the system presents the form fields where the customer is supposed to provide details on what type of website he wants to get.

```

echo "<form action=\"custom-order.php\" method=\"post\">";
echo "<table>";

echo "<tr>";
echo "<td class=\"custom\"><strong>Main Purpose</strong></td>";
echo "<td>
<input type=\"radio\" name=\"mainpurpose\" value=\"personal\">
Personal<br />
<input type=\"radio\" name=\"mainpurpose\" value=\"business\">
Business
</td>";
echo "</tr>";

```

```

echo "<tr>";
echo "<td class=\"custom\"><strong>Type of Business</strong>
(optional)</td>";
echo "<td><input type=\"text\" name=\"typeofbusiness\" ></td>";
echo "</tr>";

// Break
echo "<tr>";
echo "<td>&nbsp;</td>";
echo "<td>&nbsp;</td>";
echo "</tr>";

echo "<tr>";
echo "<td class=\"custom\"><strong>Target Reader</strong>
(optional)</td>";
echo "<td><input type=\"text\" name=\"targetreader\" ></td>";
echo "</tr>";

echo "<tr>";
echo "<td class=\"custom\"><strong>Number of Pages</strong></td>";
echo "<td><select name=\"numberofpages\">
<option value=\"0\">0 ($299)</option>
<option value=\"1\">1 ($399)</option>
<option value=\"2\">2 ($409)</option>
<option value=\"3\">3 ($419)</option>
<option value=\"4\">4 ($429)</option>
<option value=\"5\" selected>5 ($439)</option>
<option value=\"6\">6 ($449)</option>
<option value=\"7\">7 ($459)</option>
<option value=\"8\">8 ($469)</option>
<option value=\"9\">9 ($479)</option>
<option value=\"10\">10 ($489)</option>
</select></td>";
echo "</tr>";

echo "<tr>";
echo "<td class=\"custom\"></td>";
echo "<td>";
echo "Describe what your pages will be about:<br /><textarea
name=\"pagedescriptions\"></textarea><br />";
echo "</td>";
echo "</tr>";

// Break
echo "<tr>";
echo "<td>&nbsp;</td>";
echo "<td>&nbsp;</td>";
echo "</tr>";

echo "<tr>";
echo "<td class=\"custom\"><strong>Main Colors</strong></td>";
echo "<td>
<input type=\"text\" name=\"maincolor1\" ><br />
<input type=\"text\" name=\"maincolor2\" ><br />
<input type=\"text\" name=\"maincolor3\" >
</td>";
echo "</tr>";

echo "<tr>";

```

```

echo "<td class=\"custom\"><strong>Tone</strong></td>";
echo "<td>
<input type=\"radio\" name=\"tone\" value=\"light\"> Light<br />
<input type=\"radio\" name=\"tone\" value=\"dark\"> Dark
</td>";
echo "</tr>";

// Break
echo "<tr>";
echo "<td>&nbsp;&nbsp;&nbsp;</td>";
echo "<td>&nbsp;&nbsp;&nbsp;</td>";
echo "</tr>";

echo "<tr>";
echo "<td class=\"custom\"><strong>Extras</strong></td>";
echo "<td>
<input type=\"checkbox\" name=\"contactform\" value=\"yes\" > Email
contact form <span class=\"green\">$29</span><br />
<input type=\"checkbox\" name=\"googleanalytics\" value=\"yes\" >
Google Analytics (site statistics) <span class=\"green\">$29</span><br
/>
<input type=\"checkbox\" name=\"3revisions\" value=\"yes\" > 3 x
design revisions <span class=\"green\">$99</span><br />
<input type=\"checkbox\" name=\"consulting\" value=\"yes\" >
Consulting (3 hours per week, for 4 weeks) <span
class=\"green\">$129</span><br />
<input type=\"checkbox\" name=\"socialmedia\" value=\"yes\" > Social
media integration (Facebook, Twitter, Digg, etc.) <span
class=\"green\">$29</span><br />
<input type=\"checkbox\" name=\"mailinglist\" value=\"yes\" > Mailing
list <span class=\"green\">$19</span><br />
<input type=\"checkbox\" name=\"promotionalbanner\" value=\"yes\" >
Your own, branded promotional banner <span
class=\"green\">$49</span><br />
<input type=\"checkbox\" name=\"paymentsystem\" value=\"yes\" >
Instant payment system (via PayPal or credit card) <span
class=\"green\">$79</span><br />
<input type=\"checkbox\" name=\"shoppingcart\" value=\"yes\" >
Shopping cart <span class=\"green\">$249</span><br />
<input type=\"checkbox\" name=\"discussionforum\" value=\"yes\" >
Discussion forum <span class=\"green\">$49</span><br />
<input type=\"checkbox\" name=\"photoalbum\" value=\"yes\" > Photo
album <span class=\"green\">$29</span><br />
<input type=\"checkbox\" name=\"adsense\" value=\"yes\" > Google
AdSense (get paid for ad clicks) <span class=\"green\">$39</span><br
/>
</td>";
echo "</tr>";

// Break
echo "<tr>";
echo "<td>&nbsp;&nbsp;&nbsp;</td>";
echo "<td>&nbsp;&nbsp;&nbsp;</td>";
echo "</tr>";

echo "<tr>";
echo "<td class=\"custom\"><strong>Other Requests</strong></td>";
echo "<td>";
echo "<textarea name=\"otherrequests\"></textarea><br />

```

```

<small>Some requests will be priced separately on a case-to-case
basis.</small>";
echo "</td>";
echo "</tr>";

// Break
echo "<tr>";
echo "<td>&nbsp;&nbsp;&nbsp;</td>";
echo "<td>&nbsp;&nbsp;&nbsp;</td>";
echo "</tr>";

echo "<tr>";
echo "<td></td>";
echo "<td><input type=\"submit\" value=\"Continue\" /></td>";
echo "</tr>";
echo "</table>";
echo "</form>";

```

The order page then extracts the submitted data from the `$_SESSION` array, and also creates temporary variables that can be easily accessed.

```

$_SESSION["custom_mainpurpose"] = $_POST["mainpurpose"];
$_SESSION["custom_typeofbusiness"] = $_POST["typeofbusiness"];
$_SESSION["custom_targetreader"] = $_POST["targetreader"];
$_SESSION["custom_numberofpages"] = $_POST["numberofpages"];
$_SESSION["custom_pagedescriptions"] = $_POST["pagedescriptions"];
$_SESSION["custom_maincolor1"] = $_POST["maincolor1"];
$_SESSION["custom_maincolor2"] = $_POST["maincolor2"];
$_SESSION["custom_maincolor3"] = $_POST["maincolor3"];
$_SESSION["custom_tone"] = $_POST["tone"];
$_SESSION["custom_contactform"] = $_POST["contactform"];
$_SESSION["custom_googleanalytics"] = $_POST["googleanalytics"];
$_SESSION["custom_threerevisions"] = $_POST["threerevisions"];
$_SESSION["custom_consulting"] = $_POST["consulting"];
$_SESSION["custom_socialmedia"] = $_POST["socialmedia"];
$_SESSION["custom_mailinglist"] = $_POST["mailinglist"];
$_SESSION["custom_promotionalbanner"] = $_POST["promotionalbanner"];
$_SESSION["custom_paymentsystem"] = $_POST["paymentsystem"];
$_SESSION["custom_shoppingcart"] = $_POST["shoppingcart"];
$_SESSION["custom_discussionforum"] = $_POST["discussionforum"];
$_SESSION["custom_photoalbum"] = $_POST["photoalbum"];
$_SESSION["custom_adsense"] = $_POST["adsense"];
$_SESSION["custom_otherrequests"] = $_POST["otherrequests"];

$custom_mainpurpose = $_SESSION["custom_mainpurpose"];
$custom_typeofbusiness = $_SESSION["custom_typeofbusiness"];
$custom_targetreader = $_SESSION["custom_targetreader"];
$custom_numberofpages = $_SESSION["custom_numberofpages"];
$custom_pagedescriptions = $_SESSION["custom_pagedescriptions"];
$custom_maincolor1 = $_SESSION["custom_maincolor1"];
$custom_maincolor2 = $_SESSION["custom_maincolor2"];
$custom_maincolor3 = $_SESSION["custom_maincolor3"]; $custom_tone =
$_SESSION["custom_tone"]; $custom_contactform =
$_SESSION["custom_contactform"]; $custom_googleanalytics =
$_SESSION["custom_googleanalytics"]; $custom_threerevisions =
$_SESSION["custom_threerevisions"]; $custom_consulting =
$_SESSION["custom_consulting"]; $custom_socialmedia =
$_SESSION["custom_socialmedia"]; $custom_mailinglist =
$_SESSION["custom_mailinglist"]; $custom_promotionalbanner =
$_SESSION["custom_promotionalbanner"]; $custom_paymentsystem =

```

```

$_SESSION["custom_paymentsystem"]; $custom_shoppingcart =
$_SESSION["custom_shoppingcart"]; $custom_discussionforum =
$_SESSION["custom_discussionforum"]; $custom_photoalbum =
$_SESSION["custom_photoalbum"]; $custom_adsense =
$_SESSION["custom_adsense"]; $custom_otherrequests =
$_SESSION["custom_otherrequests"];

```

The following piece of code is placed on this page and on every subsequent page in order to prevent people from accessing these pages without having submitted valid information to the forms.

```

// Validate continuity
if ($_POST["numberofpages"] == NULL) {
    echo "<p>Please start over.</p>";
} else {

```

Next, the system calculates the cost, first by calculating the base price depending on the number of pages ordered, and then with the extras (where appropriate). After that, it stores the total cost into the `$_SESSION` array for later use.

```

// Set base price
if ($custom_numberofpages == 0) {
    $totalcost = 299;
}
elseif ($custom_numberofpages == 1) {
    $totalcost = 399;
} elseif ($custom_numberofpages == 2) {
    $totalcost = 409;
} elseif ($custom_numberofpages == 3) {
    $totalcost = 419;
} elseif ($custom_numberofpages == 4) {
    $totalcost = 429;
} elseif ($custom_numberofpages == 5) {
    $totalcost = 439;
} elseif ($custom_numberofpages == 6) {
    $totalcost = 449;
} elseif ($custom_numberofpages == 7) {
    $totalcost = 459;
} elseif ($custom_numberofpages == 8) {
    $totalcost = 469;
} elseif ($custom_numberofpages == 9) {
    $totalcost = 479;
} elseif ($custom_numberofpages == 10) {
    $totalcost = 489;
}

// Calculate total cost
if ($custom_contactform == "yes") {
    $totalcost = $totalcost + 29;
}
if ($custom_googleanalytics == "yes") {
    $totalcost = $totalcost + 29;
}
if ($custom_threerevisions == "yes") {
    $totalcost = $totalcost + 99;
}
if ($custom_consulting == "yes") {
    $totalcost = $totalcost + 129;
}
if ($custom_socialmedia == "yes") {

```

```

        $totalcost = $totalcost + 29;
    }
    if ($custom_mailinglist == "yes") {
        $totalcost = $totalcost + 19;
    }
    if ($custom_promotionalbanner == "yes") {
        $totalcost = $totalcost + 49;
    }
    if ($custom_paymentsystem == "yes") {
        $totalcost = $totalcost + 79;
    }
    if ($custom_shoppingcart == "yes") {
        $totalcost = $totalcost + 249;
    }
    if ($custom_discussionforum == "yes") {
        $totalcost = $totalcost + 49;
    }
    if ($custom_photoalbum == "yes") {
        $totalcost = $totalcost + 29;
    }
    if ($custom_adsense == "yes") {
        $totalcost = $totalcost + 39;
    }
}

$_SESSION["custom_totalcost"] = $totalcost;

```

When the calculations are done, the system prompts the customer to fill out either the registration form or the login form. Since the registration form was covered in the previous section, the login form will be explained this time.

The login form bypasses the registration phase if the customer has already purchased something from the site before. When a customer has ordered a product and registered an account, he or she will be able to make future orders without having to input all the personal details again. A simple check that requires the email address and password allows the customer to place an order instantly.

```

echo "<form action=\"custom-login.php\" method=\"post\">";
echo "<table>";
echo "<tr>";
echo "<td class=\"first\"><strong>Email</strong></td>";
echo "<td><input type=\"text\" name=\"email\" ></td>";
echo "</tr>";
echo "<tr>";
echo "<td class=\"first\"><strong>Password</strong></td>";
echo "<td><input type=\"password\" name=\"password\" ></td>";
echo "</tr>";
echo "<tr>";
echo "<td>&nbsp;</td>";
echo "<td>&nbsp;</td>";
echo "</tr>";
echo "<tr>";
echo "<td></td>";
echo "<td><input type=\"submit\" value=\"Confirm Order\" /></td>";
echo "</tr>";
echo "</table>";
echo "</form>";

```

When the login items are sent to the system, it checks that they are valid.

```
// Data from form
```

```

$email = $_POST["email"];
$password = $_POST["password"];

// Retrieve customer info from database
$query_customerinfo = "SELECT * FROM customer WHERE email = '$email'";
$get_customerinfo = mysql_query($query_customerinfo, $link);

// Check if customer exists
if (mysql_num_rows($get_customerinfo) == 0) {
    $customer_exists = FALSE;
} else {
    $customer_exists = TRUE;
}

if ($customer_exists == FALSE) {
    echo "<p>Customer does not exist.</p><form><input
type=\"button\" value=\"Go Back\" onClick=\"history.go(-1);return
true;\" /></form>";
} else {

// Check if account and password match
$retrieved_password = mysql_result($get_customerinfo, 0, "password");
if ($password == $retrieved_password) {
    $password_ok = TRUE;
}
if ($password_ok == FALSE) {
    echo "<p>Wrong password.</p><form><input type=\"button\"
value=\"Go Back\" onClick=\"history.go(-1);return true;\" /></form>";
} else {

```

A valid login entry will then retrieve the customer information from the database and order data from the previous pages.

```

$customerid = mysql_result($get_customerinfo, 0, "customerid");
$firstname = mysql_result($get_customerinfo, 0, "firstname");
$lastname = mysql_result($get_customerinfo, 0, "lastname"); $address1 =
mysql_result($get_customerinfo, 0, "address1"); $address2 =
mysql_result($get_customerinfo, 0, "address2"); $postalcode =
mysql_result($get_customerinfo, 0, "postalcode"); $city =
mysql_result($get_customerinfo, 0, "city"); $country =
mysql_result($get_customerinfo, 0, "country"); $phonenumber =
mysql_result($get_customerinfo, 0, "phonenumber"); $gender =
mysql_result($get_customerinfo, 0, "gender"); // Order variables
$custom_mainpurpose = $_SESSION["custom_mainpurpose"];
$custom_typeofbusiness = $_SESSION["custom_typeofbusiness"];
$custom_targetreader = $_SESSION["custom_targetreader"];
$custom_numberofpages = $_SESSION["custom_numberofpages"];
$custom_pagedescriptions = $_SESSION["custom_pagedescriptions"];
$custom_maincolor1 = $_SESSION["custom_maincolor1"];
$custom_maincolor2 = $_SESSION["custom_maincolor2"];
$custom_maincolor3 = $_SESSION["custom_maincolor3"]; $custom_tone =
$_SESSION["custom_tone"]; $custom_contactform =
$_SESSION["custom_contactform"]; $custom_googleanalytics =
$_SESSION["custom_googleanalytics"]; $custom_threerevisions =
$_SESSION["custom_threerevisions"]; $custom_consulting =
$_SESSION["custom_consulting"]; $custom_socialmedia =
$_SESSION["custom_socialmedia"]; $custom_mailinglist =
$_SESSION["custom_mailinglist"]; $custom_promotionalbanner =
$_SESSION["custom_promotionalbanner"]; $custom_paymentsystem =
$_SESSION["custom_paymentsystem"]; $custom_shoppingcart =

```



```

$_SESSION["custom_shoppingcart"]; $custom_discussionforum =
$_SESSION["custom_discussionforum"]; $custom_photoalbum =
$_SESSION["custom_photoalbum"]; $custom_adsense =
$_SESSION["custom_adsense"]; $custom_otherrequests =
$_SESSION["custom_otherrequests"]; $custom_totalcost =
$_SESSION["custom_totalcost"];

```

Then finally the order will be stored into the database.

```

$insert_order = "INSERT INTO custompackorder (customerid, mainpurpose,
typeofbusiness, targetreader, numberofpages, pagedescriptions,
maincolor1, maincolor2, maincolor3, tone, contactform,
googleanalytics, threerevisions, consulting, socialmedia, mailinglist,
promotionalbanner, paymentsystem, shoppingcart, discussionforum,
photoalbum, adsense, otherrequests, totalcost)
VALUES ('$customerid', '$custom_mainpurpose',
'$custom_typeofbusiness', '$custom_targetreader',
'$custom_numberofpages', '$custom_pagedescriptions',
'$custom_maincolor1', '$custom_maincolor2', '$custom_maincolor3',
'$custom_tone', '$custom_contactform', '$custom_googleanalytics',
'$custom_threerevisions', '$custom_consulting', '$custom_socialmedia',
'$custom_mailinglist', '$custom_promotionalbanner',
'$custom_paymentsystem', '$custom_shoppingcart',
'$custom_discussionforum', '$custom_photoalbum', '$custom_adsense',
'$custom_otherrequests', '$custom_totalcost')";
mysql_query($insert_order, $link) or die("Error storing order.");
$_SESSION["neworderid"] = mysql_insert_id();
$neworderid = $_SESSION["neworderid"];

```

And the data will be shown on the website for confirmation.

```

echo "<table>";

echo "<tr>";
echo "<td class=\"custom\"><strong>Main Purpose</strong></td>";
echo "<td class=\"custom\">$custom_mainpurpose</td>";
echo "</tr>";

echo "<tr>";
echo "<td class=\"custom\"><strong>Type of Business</strong></td>";
echo "<td class=\"custom\">$custom_typeofbusiness</td>";
echo "</tr>";

// Break
echo "<tr>";
echo "<td class=\"custom\">&nbsp;</td>";
echo "<td class=\"custom\">&nbsp;</td>";
echo "</tr>";

echo "<tr>";
echo "<td class=\"custom\"><strong>Target Reader</strong></td>";
echo "<td class=\"custom\">$custom_target_reader</td>";
echo "</tr>";

echo "<tr>";
echo "<td class=\"custom\"><strong>Number of Pages</strong></td>";
echo "<td class=\"custom\">$custom_numberofpages</td>";
echo "</tr>";

// Break

```

```

echo "<tr>";
echo "<td class=\"custom\">&nbsp;</td>";
echo "<td class=\"custom\">&nbsp;</td>";
echo "</tr>";

echo "<tr>";
echo "<td class=\"custom\"><strong>Page Descriptions</strong></td>";
echo "<td class=\"custom\">${custom_pagedescriptions}</td>";
echo "</tr>";

echo "<tr>";
echo "<td class=\"custom\"><strong>Main Colors</strong></td>";
echo "<td class=\"custom\">${custom_maincolor1}, ${custom_maincolor2},
${custom_maincolor3}</td>";
echo "</tr>";

echo "<tr>";
echo "<td class=\"custom\"><strong>Tone</strong></td>";
echo "<td class=\"custom\">${custom_tone}</td>";
echo "</tr>";

// Break
echo "<tr>";
echo "<td class=\"custom\">&nbsp;</td>";
echo "<td class=\"custom\">&nbsp;</td>";
echo "</tr>";

echo "<tr>";
echo "<td class=\"custom\"><strong>Email contact form</strong></td>";
echo "<td class=\"custom\">${custom_contactform}</td>";
echo "</tr>";

echo "<tr>";
echo "<td class=\"custom\"><strong>Google Analytics</strong></td>";
echo "<td class=\"custom\">${custom_googleanalytics}</td>";
echo "</tr>";

echo "<tr>";
echo "<td class=\"custom\"><strong>3 x design
revisions</strong></td>";
echo "<td class=\"custom\">${custom_threerevisions}</td>";
echo "</tr>";

echo "<tr>";
echo "<td class=\"custom\"><strong>Consulting (3 hours per week, for 4
weeks)</strong></td>";
echo "<td class=\"custom\">${custom_consulting}</td>";
echo "</tr>";

echo "<tr>";
echo "<td class=\"custom\"><strong>Social media integration (Facebook,
Twitter, Digg, etc.)</strong></td>";
echo "<td class=\"custom\">${custom_socialmedia}</td>";
echo "</tr>";

echo "<tr>";
echo "<td class=\"custom\"><strong>Mailing list</strong></td>";
echo "<td class=\"custom\">${custom_mailinglist}</td>";
echo "</tr>";

```

```

echo "<tr>";
echo "<td class=\"custom\"><strong>Your own, branded promotional
banner</strong></td>";
echo "<td class=\"custom\">$custom_promotionalbanner</td>";
echo "</tr>";

echo "<tr>";
echo "<td class=\"custom\"><strong>Instant payment
system</strong></td>";
echo "<td class=\"custom\">$custom_paymentsystem</td>";
echo "</tr>";

echo "<tr>";
echo "<td class=\"custom\"><strong>Shopping cart</strong></td>";
echo "<td class=\"custom\">$custom_shoppingcart</td>";
echo "</tr>";

echo "<tr>";
echo "<td class=\"custom\"><strong>Discussion forum</strong></td>";
echo "<td class=\"custom\">$custom_discussionforum</td>";
echo "</tr>";

echo "<tr>";
echo "<td class=\"custom\"><strong>Photo album</strong></td>";
echo "<td class=\"custom\">$custom_photoalbum</td>";
echo "</tr>";

echo "<tr>";
echo "<td class=\"custom\"><strong>Google AdSense</strong></td>";
echo "<td class=\"custom\">$custom_adsense</td>";
echo "</tr>";

// Break
echo "<tr>";
echo "<td class=\"custom\">&nbsp;</td>";
echo "<td class=\"custom\">&nbsp;</td>";
echo "</tr>";

echo "<tr>";
echo "<td class=\"custom\"><strong>Other requests</strong></td>";
echo "<td class=\"custom\">$custom_otherrequests</td>";
echo "</tr>";

// Break
echo "<tr>";
echo "<td class=\"custom\">&nbsp;</td>";
echo "<td class=\"custom\">&nbsp;</td>";
echo "</tr>";

echo "<tr>";
echo "<td class=\"custom\"><strong>Total Cost</strong></td>";
echo "<td class=\"custom\">$$custom_totalcost (USD)</td>";
echo "</tr>";

echo "</table>";

echo "<br />";

echo "<h1>Your Information</h1>";
echo "<table>";

```

```

echo "<tr>";
echo "<td class=\"custom\"><strong>First Name</strong></td>";
echo "<td class=\"custom\">${firstname}</td>";
echo "</tr>";
echo "<tr>";
echo "<td class=\"custom\"><strong>Last Name</strong></td>";
echo "<td class=\"custom\">${lastname}</td>";
echo "</tr>";
echo "<tr>";
echo "<td class=\"custom\"><strong>Address 1</strong></td>";
echo "<td class=\"custom\">${address1}</td>";
echo "</tr>";
echo "<tr>";
echo "<td class=\"custom\"><strong>Address 2</strong></td>";
echo "<td class=\"custom\">${address2}</td>";
echo "</tr>";
echo "<tr>";
echo "<td class=\"custom\"><strong>Postal Code</strong></td>";
echo "<td class=\"custom\">${postalcode}</td>";
echo "</tr>";
echo "<tr>";
echo "<td class=\"custom\"><strong>City</strong></td>";
echo "<td class=\"custom\">${city}</td>";
echo "</tr>";
echo "<tr>";
echo "<td class=\"custom\"><strong>Country</strong></td>";
echo "<td class=\"custom\">${country}</td>";
echo "</tr>";
echo "<tr>";
echo "<td class=\"custom\"><strong>Phone Number</strong></td>";
echo "<td class=\"custom\">${phonenumber}</td>";
echo "</tr>";
echo "<tr>";
echo "<td class=\"custom\"><strong>Gender</strong></td>";
echo "<td class=\"custom\">${gender}</td>";
echo "</tr>";
echo "<tr>";
echo "<td class=\"custom\"><strong>Email</strong></td>";
echo "<td class=\"custom\">${email}</td>";
echo "</tr>";
echo "<tr>";
echo "<td class=\"custom\"><strong>Password</strong></td>";
echo "<td class=\"custom\">${password}</td>";
echo "</tr>";
echo "</table>";

```

As with pre-compiled orders, the order is now placed so the customer can safely close the browser if all the information is correct. If he is unhappy with the order, the order can still be canceled on this page by clicking on the *Cancel Order* button, which has the same function as was explained earlier.