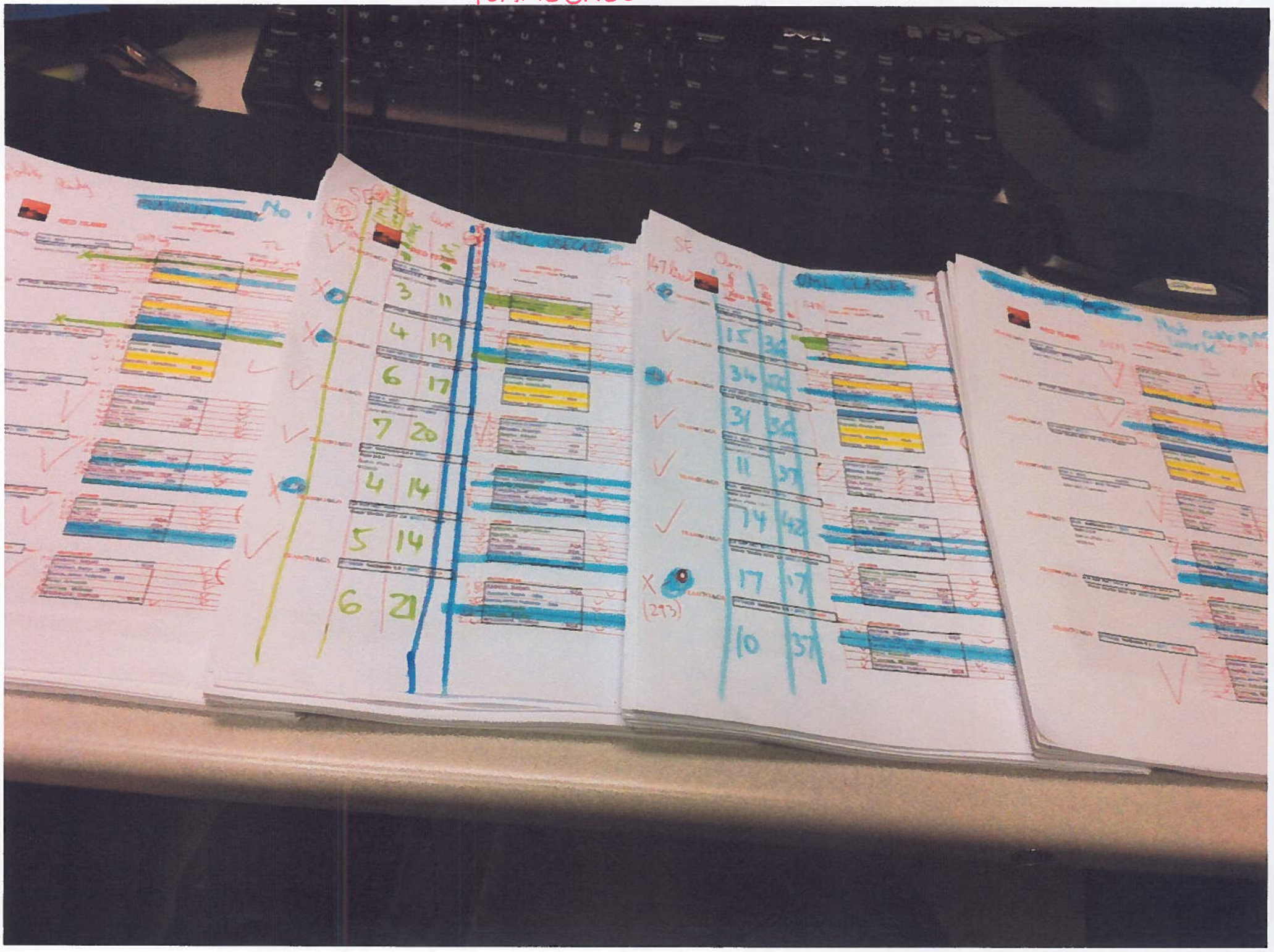
TEAMS BAGS


3 11
4 19
6 17
7 20
4 14
5 14
6 21
15 34
34 42
31 32
11 37
14 42
17 17
10 37
(293)

100

TEAM5OAGS

Feasibility Study

Liang Jaradi | No work assigned!

Version 2.0

Great SVM work!

Signature Block

COSC 4351	Name	Signature	Date
SE Team Leader	Ryan Lee		2/6/2013
SE SQA	Christopher Cruz		2/6/2013
SE SQA	Muhammad Naviwala		2/6/2013

DOCUMENT CONTROL

CHANGE HISTORY

TLs entries (assigned work and due dates) before releasing to the team

Revision	Name	Due Date	Description
1.1	Joel Loucks	02/01/2013	Complete Document
1.2	Christopher Berkowitz	02/01/2013	Complete Document
1.3	Gabriel Ohlson	02/01/2013	Complete Document
1.X	Christopher Cruz	02/05/2013	Review Document
1.Y	Muhammad Naviwala	02/05/2013	Review Document
1.Z	Ryan Lee	02/05/2013	Refine and Perform Final Review Document

TMs entries when they completed their work

Revision	Name	**Completed Date**	Description
1.A	Gabriel Ohlson	01/20/2013	Added links, screen shots and technology used.
1.B	Joel Loucks	01/20/2013	Added cost estimates for each website and titles to distinguish website screen shots from cost estimate screenshots.
1.C	Christopher Cruz	01/21/2013	Reviewed document, looks good so far.
1.D	Chris Berkowitz	01/22/2013	Added new website, link, multiple SS, information required, and costs.
1.E	Christopher Cruz	01/23/2013	Corrected some grammar and added some points.
1.F	Muhammad Naviwala	01/24/2013	Modified the formatting a little bit. There is still inconsistency. Additionally, Interface column not filled out.
1.G	Muhammad Naviwala	01/30/2013	The document looks good, just the "interface" part and pros and cons have to be completed.
1.H	Chris Berkowitz	01/30/2013	Fixed spelling error.
1.I	Chris Berkowitz	01/30/2013	Added section for Interface on Sharon Norman Photography section.
1.J	Chris Berkowitz	01/30/2013	Added Section at the bottom to

Liang? Jared:

SVM

			compare the websites. Also, pointing out the nicer sites to the lesser sites.
1.K	Ryan Lee	01/30/2013	Refined feasibility study document. Added intro and case study paragraphs. Refined screenshot captions. Will continue later.
1.L	Christopher Cruz	1/30/13	Elaborated many captions, removed an incorrect caption. Checked grammar and punctuation, flow. Last webpage needs to be uniform with the rest of the layout of the document. Added interfaces for the webpages, still missing the last one.
1.M	Berkowitz, Chris	1/30/13	Changed all SS to Screenshot.
1.N	Christopher Cruz	1/31/2013	Elaborated on comments for the last page, deleted duplicated screen shot. Added interface.
1.O	Ryan Lee	02/01/2013	Added last two paragraphs to the feasibility study. Also added refinements to Case study 4 example. Highlighted paragraph needs revision.
1.P	Christopher Cruz	02/01/2013	Changed a paragraph.
1.Q	Christopher Cruz	02/05/2013	Final Review done, looks great.
1.R	Muhammad Naviwala	02/05/2013	Final review done. The document fulfills all the requirement. Looks good!
1.?	Ryan Lee	02/05/2013	Team leader final approval for Red Deliverables.

TLs entry before RED TEAMS DELIVERABLES

Revision	Name	Due Date	Description
2.0	Ryan Lee	02/05/2013	I changed Version to 2.0

DOCUMENT STORAGE

This file is stored in SVN at http://limi.cs.uh.edu/COSC4351/team5/TEAM PROJECT DELIVERABLES/**Feasibility Study.doc**.

REPLACE *Team Project* with your TEAM5OAGS all uppercase one word

Feasibility study content hints:

This document contains the information about the feasibility of the ARTSY.COM that your team is going to OOA, OOD, and OO Implement.

Find existing similar software products:

- get of link,
- get screenshots;
- get technology,
- get interface,
- get cost,

Ideally 3 of them, and then show good ones versus bad ones, and how they solved some of the technical issues, such as live chat, security.

In the process start PROTOTYPING your ARTSY.COM and decide on the WEB SITE design, each web page design, colors, pictures, buttons, etc. There should be actual work on your team project while investigating similar systems.

Introduction

TEAM 5 is a software development team that is planning to create a website for an online artwork gallery company. The company specializes in the acquisition and selling of contemporary European and North American fine art, such as lithographs, prints, paintings, and photographs. The website to be constructed will display all of the art works that the company is selling and allow customers to make online purchases of those works. The website will also contain features that will allow customers and merchants to communicate with the employees of the company, sell their artwork to the company, and view testimonials left by previous customers.

The purpose of this report is to examine the feasibility of developing and implementing the website that is to be TEAM5OAGS. In this report, the team analyzes several examples of websites that specialize in the selling of online artworks by providing their respective URLs, screenshots, technologies used (such as the web development framework and scripting languages), user interface designs, and estimated costs. Using these examples acquired, this report will determine if the solution (in this case the establishment of the online art gallery website) is feasible and appropriate. It will also provide estimated financial and time costs to develop and implement the solution.

Analysis of Examples

This report will be focus on four case studies. Each case study will focus on a particular website and will have an overview table that will list the website's name, URL, technologies used, interface description, and estimated costs. Screenshots of various web pages will also be included for greater emphasis on the content details each page presents and for preliminary designs that may be incorporated in the solution to be developed.

Case Study 1	
Website Name	UGallery
URL	http://www.ugallery.com
Technologies Used	ASP.NET on a server running Microsoft-IIS 7.0 with JavaScript
Interface Description	The website implements a four column grid using HTML and CSS for styling. There are a couple of "call to actions," which are areas that are highlighted for the user to interact with. Examples are the contest for the $500 in art and the chat feature.
Estimated Cost*	$3,000.00 for an offshore firm, $13,000.00+ for a professional firm

Screenshots of various web pages of UGallery.com are provided below.

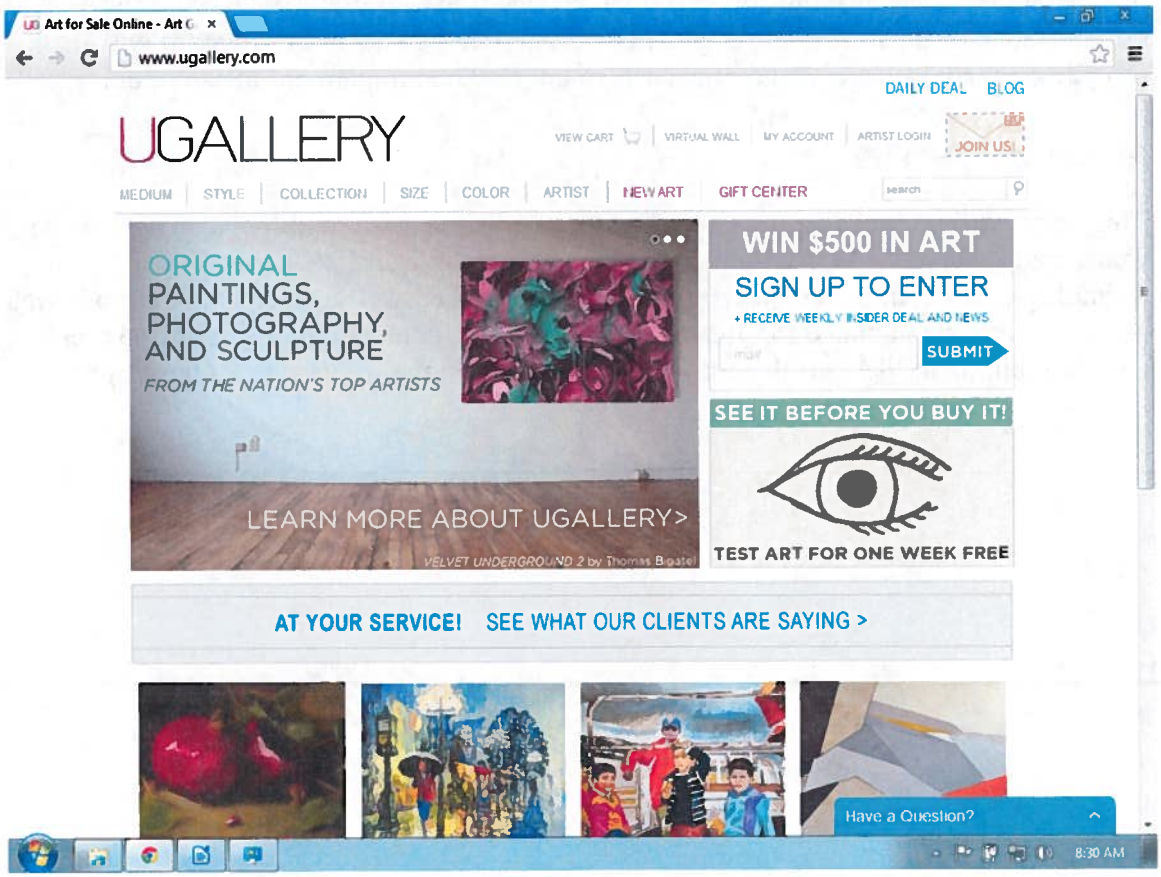


Figure 1.1: This is the home page screenshot of UGallery.com.

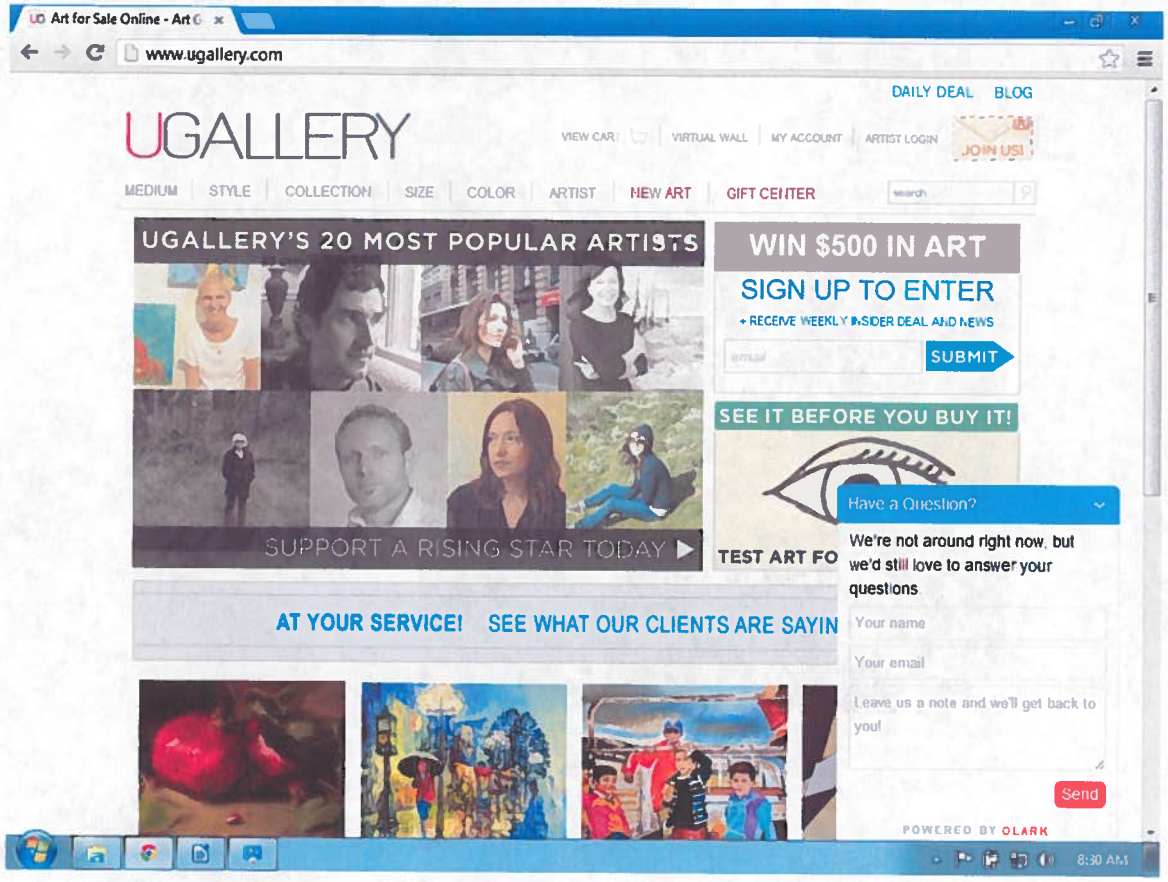


Figure 1.2: This screenshot shows a view of a live chat feature that enables the user to communicate with an online representative. If the online representative is not available, then automated responses are displayed to the customer. The automated responses are based on predefined questions. The user has the option to leave a message (email) and have a representative get back to them at a later date.

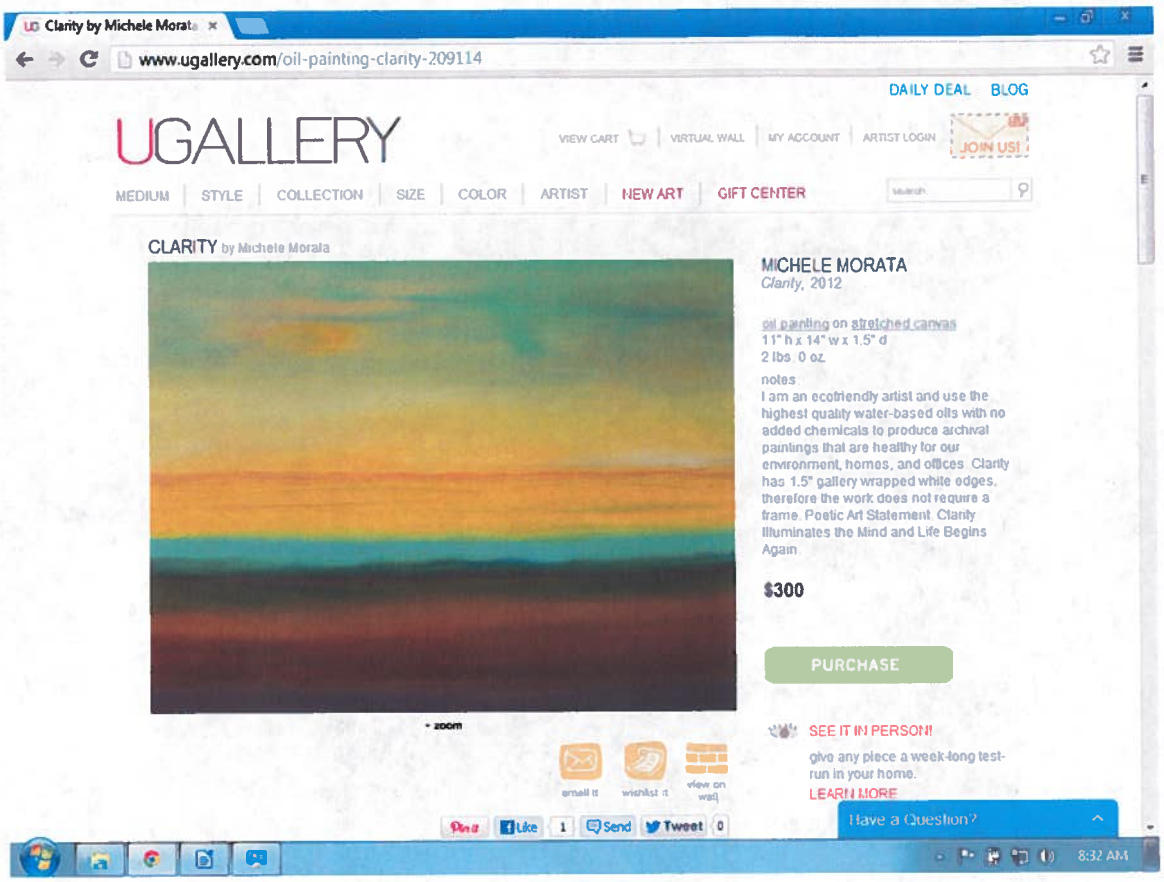


Figure 1.3: This is a screenshot of an artwork on sale. It has the sale price and name of the artist as well as a brief description of the artwork. There is also an option to purchase it or give it a "trial" run for a week. This page also has social network integration with Pinterest, Facebook, and Twitter.

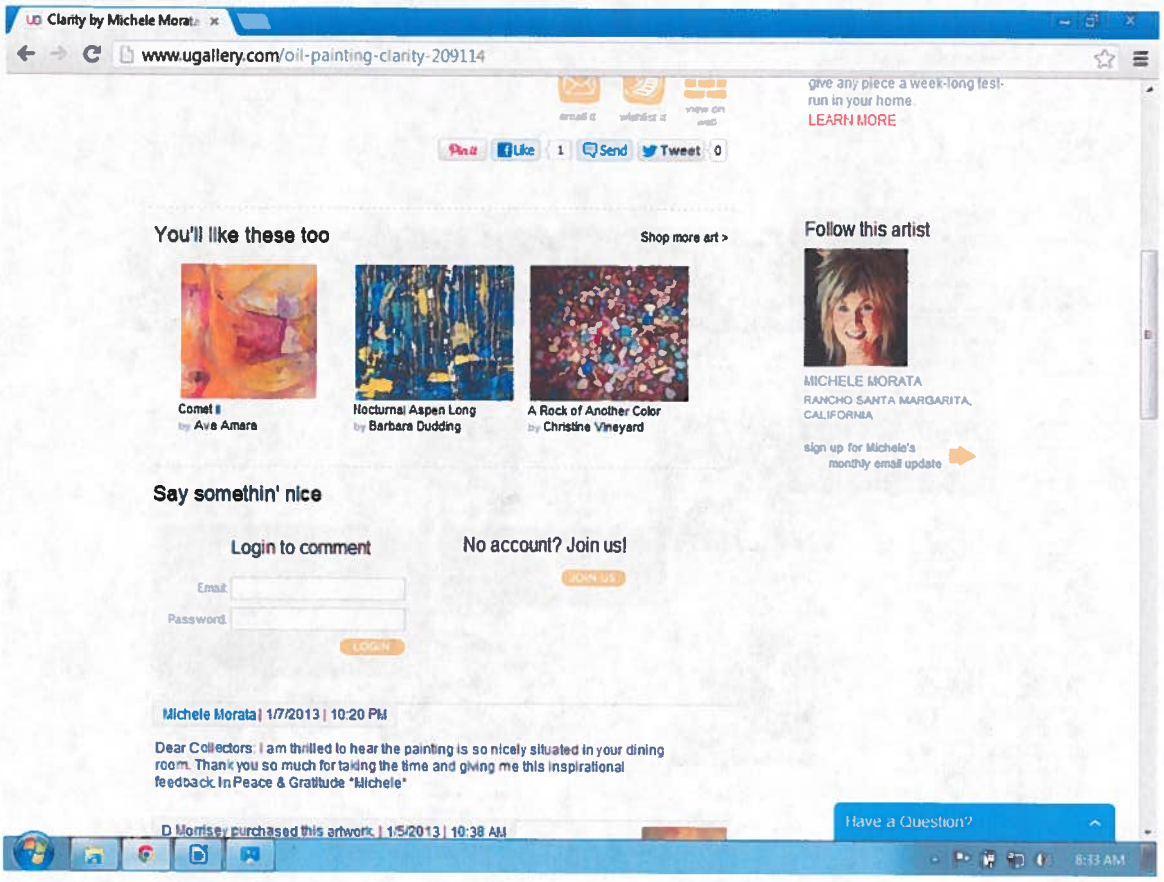


Figure 1.4: Scrolling down on the same web page shown in the previous image, a section of artworks related to the featured piece is visible. Also, there is a login feature that allows customers who have made a purchase of the featured artwork to leave comments.

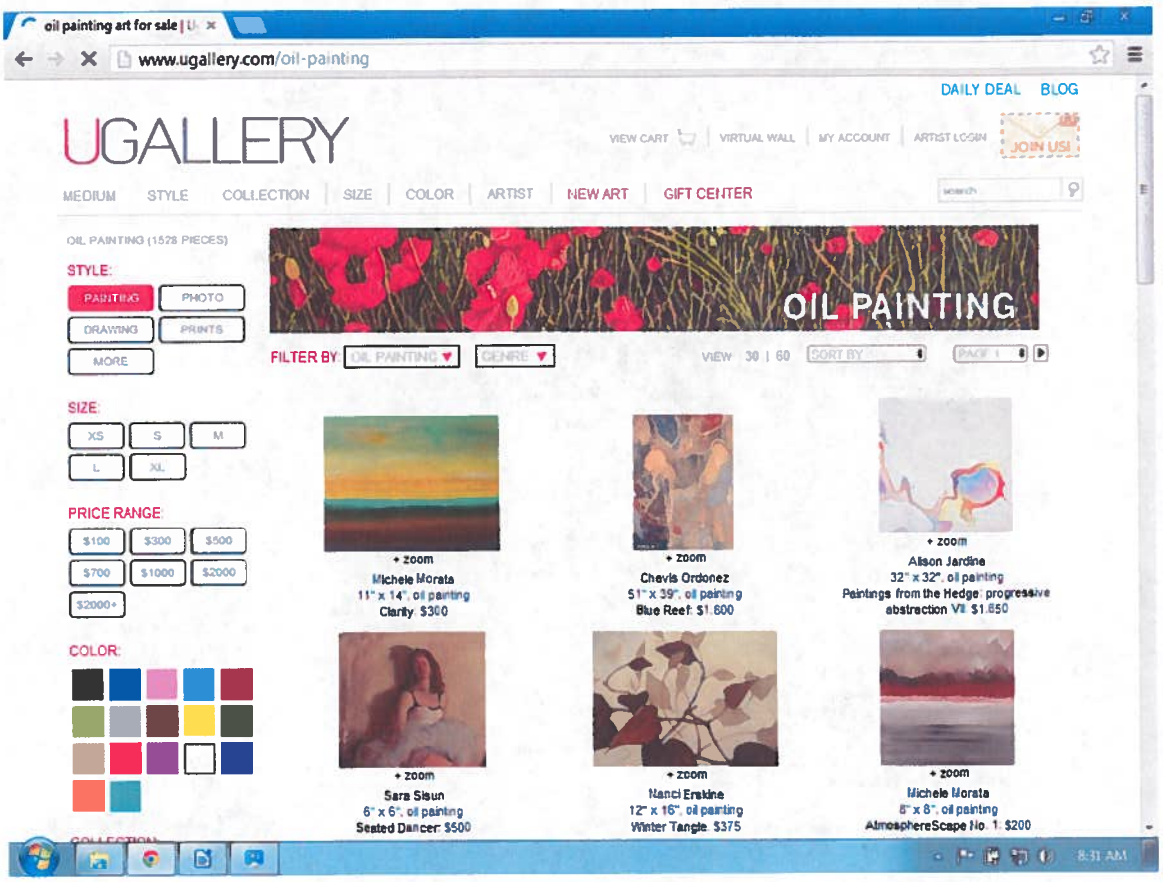


Figure 1.5: This screenshot shows the viewer a gallery view of all artworks by type, size, and price range. Art style and genre filters are available to narrow down search results. Also, the viewer can sort the results returned by using the dropdown menu labeled "sort by."

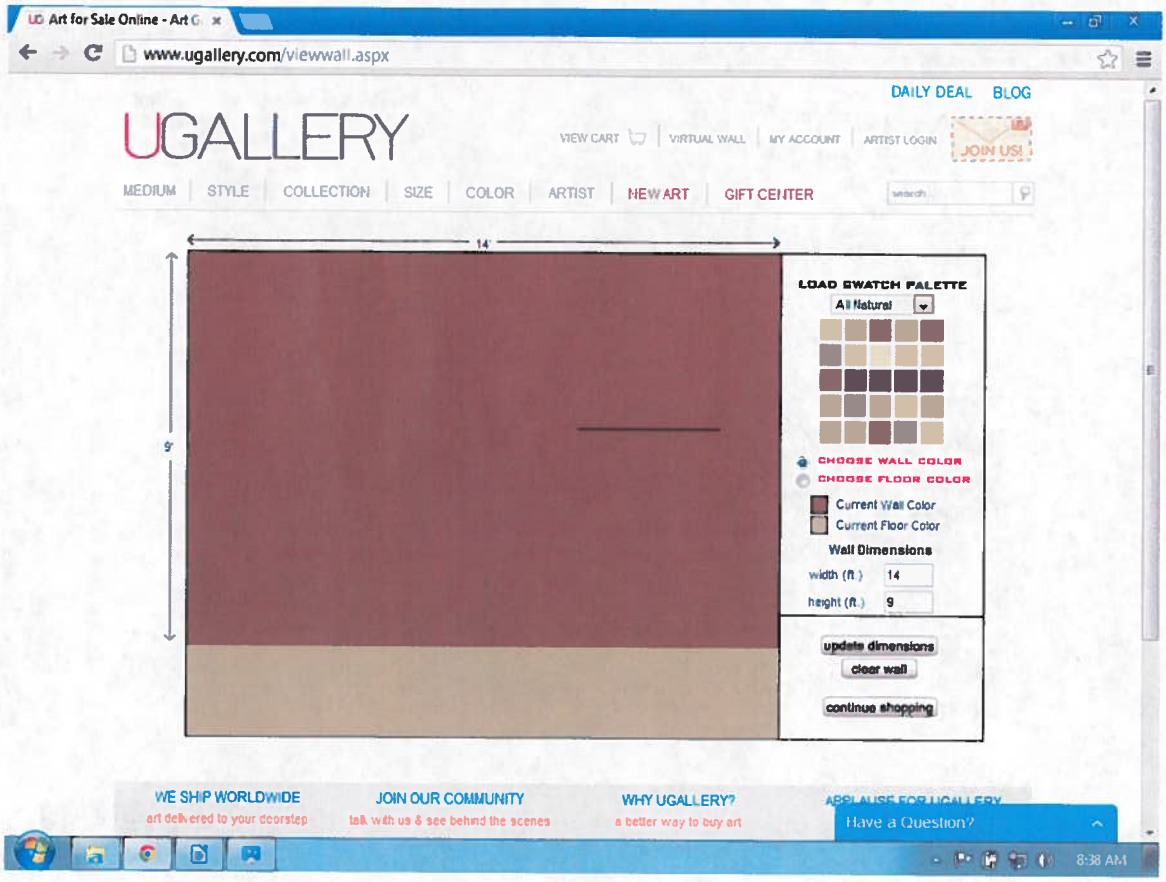


Figure 1.6: This screenshot presents the viewer an opportunity to view what their artworks in their cart would look like given a set of colors for the wall and floor. Also, the dimensions of the wall can be modified by the user.

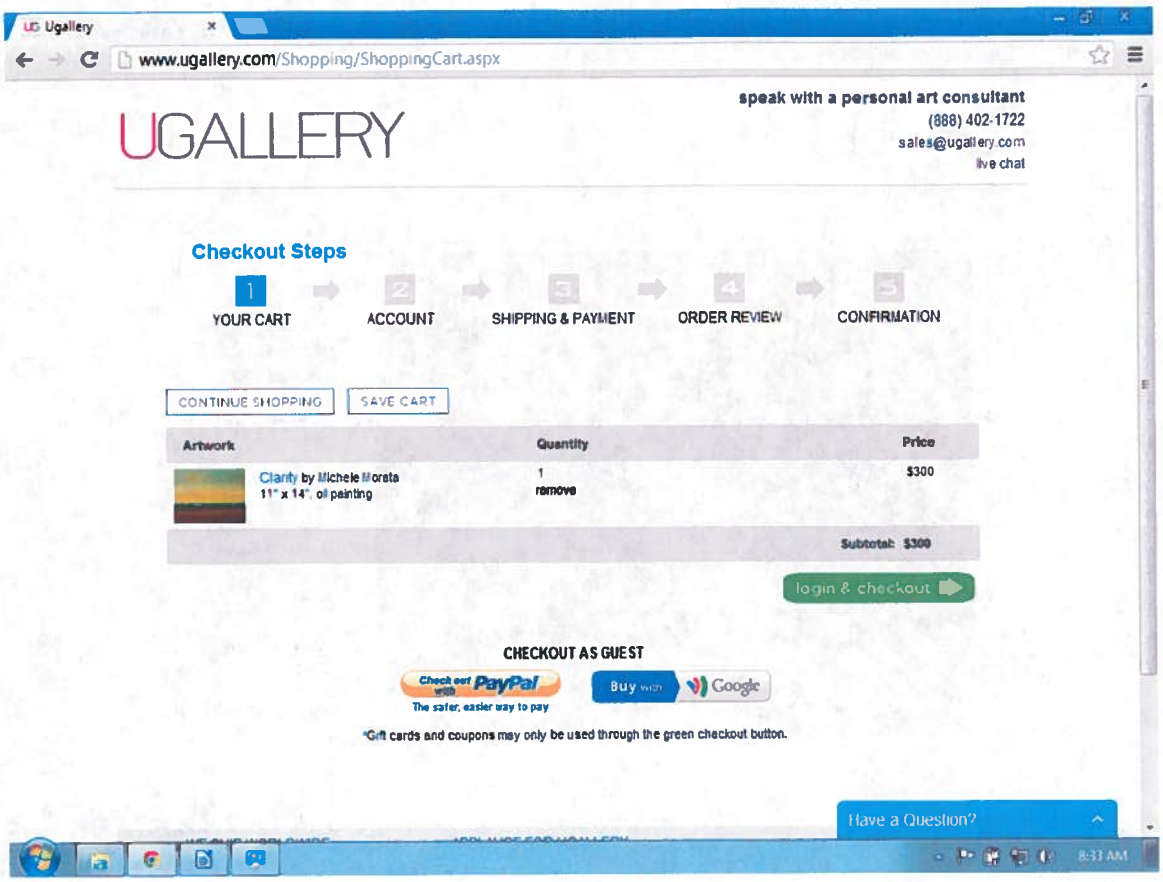


Figure 1.7: This is a screenshot that details the checkout procedures for the website. The website has a variety of ecommerce options: PayPal and Google Checkout.

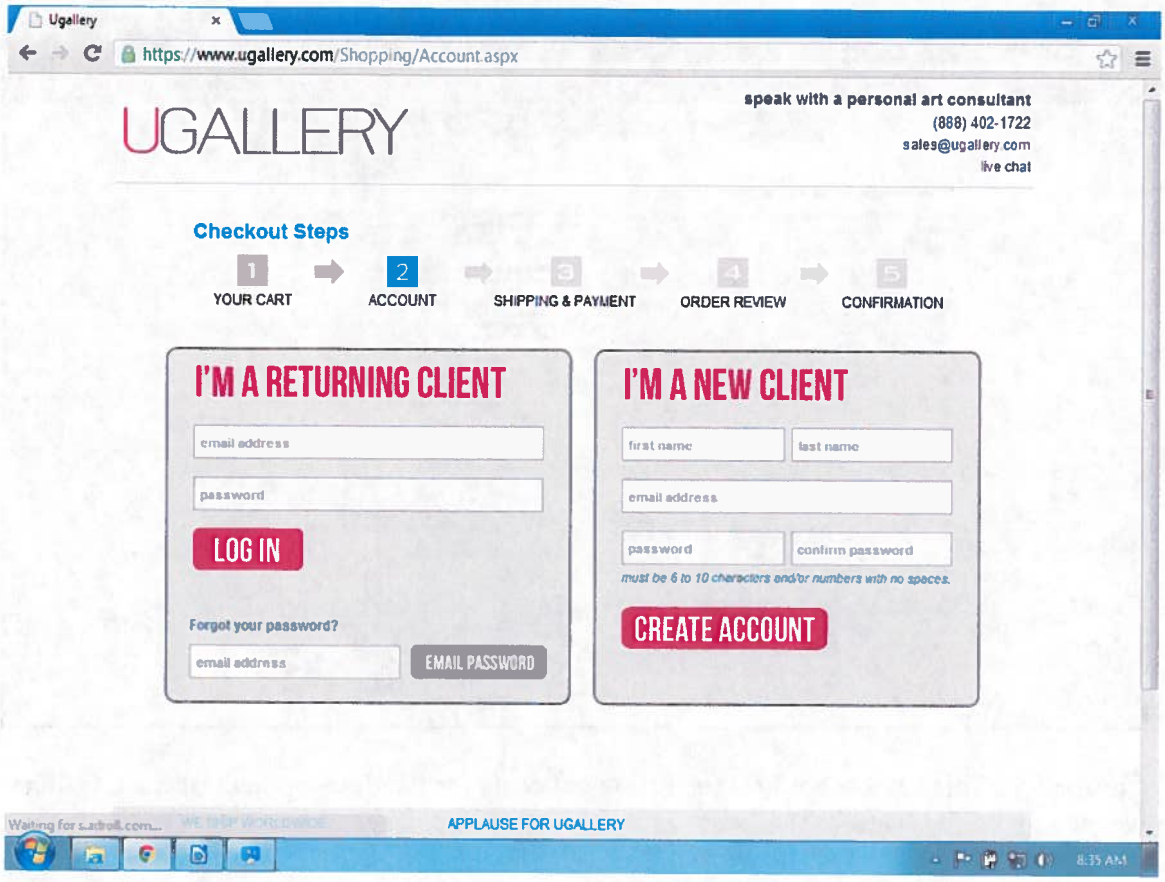


Figure 1.8: This is a screenshot for the login page. The user can create a new account or using an existing one to log in. If a password is forgotten, a password reset request can be sent from here as well.

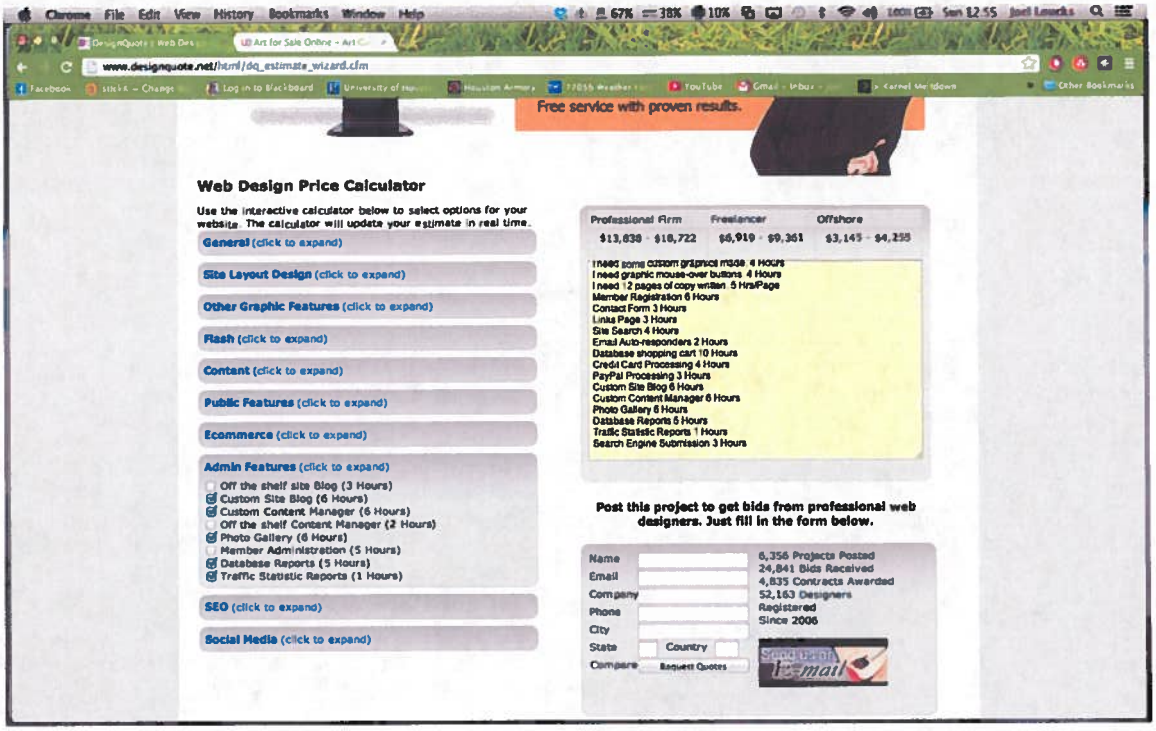


Figure 1.9: This screenshot lists the estimated costs for the development of the UGallery website given the features that were existent on the web pages.

Case Study 1 Summary

Overall, UGallery presents the viewer with a user-friendly interface. Perhaps the most notable feature that stands out in UGallery is the option that allows the viewer to see what an artwork would look like when hung on a particular wall.

Case Study 2	
Website Name	Artmajeur
URL	http://www.artmajeur.com/en/
Technologies Used	PHP 5.3.10 running on Ubuntu 3.4 with JavaScript
Interface Description	This website implements its user interface and navigation with thumbnails. Other pages also uses pagination, which informs the user that there are multiple pages of thumbnails.
Estimated Cost*	$3,000.00 for an offshore firm, $13,000.00+ for a professional firm

Screenshots of various web pages of ArtMajeur.com are provided below.

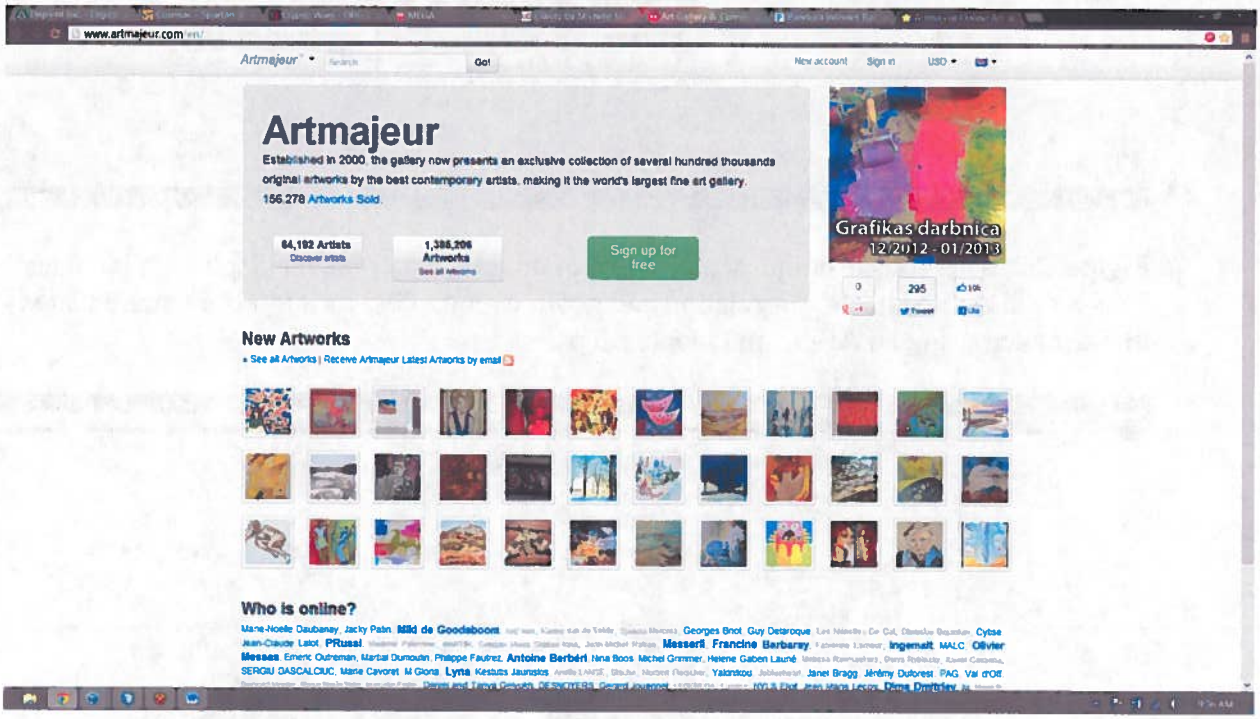


Figure 2.1: This is the screenshot for the home page of artmajeur.com/en/. There are thumbnails that show all the new artworks that have been recently added to the site. You can navigate to a list of artists, artworks, and sign up for an account. There is also integration with social networks: Facebook, Pinterest, and Twitter.

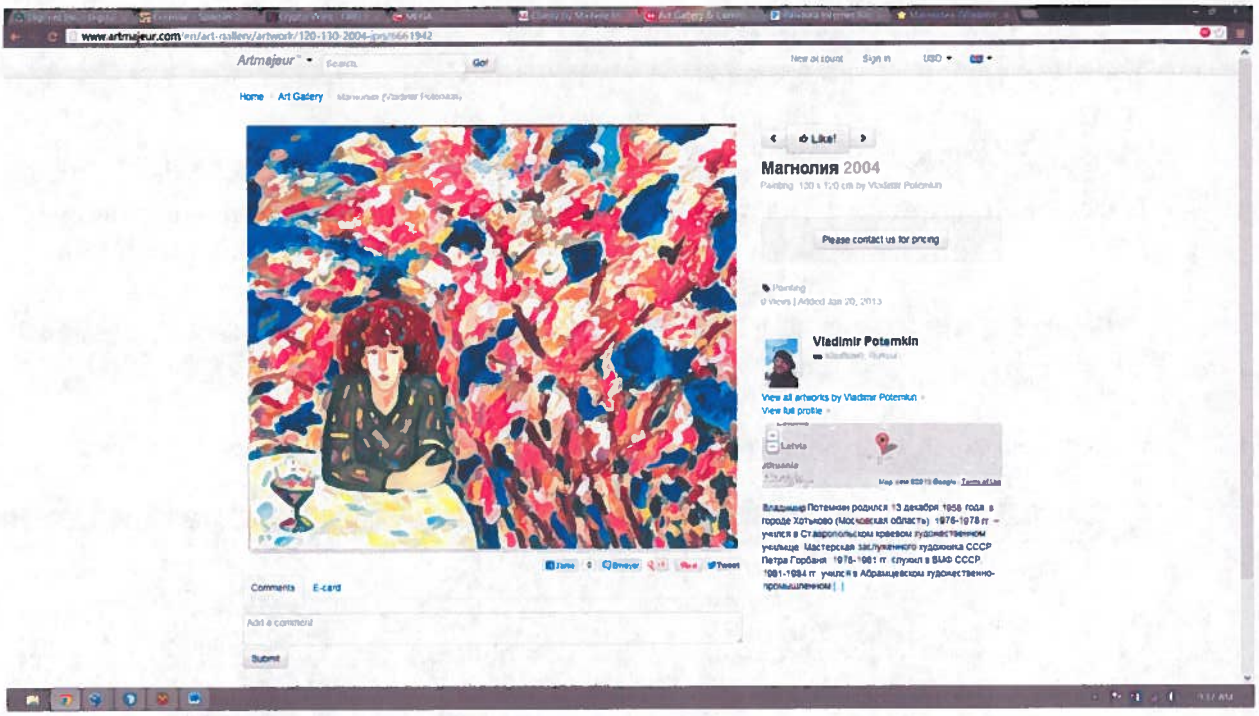


Figure 2.2: This screenshot presents a view of an artwork that is on display. It includes details such as the artist's information and geo-location. Geo-location is a feature that is implemented using an API from Google Maps.

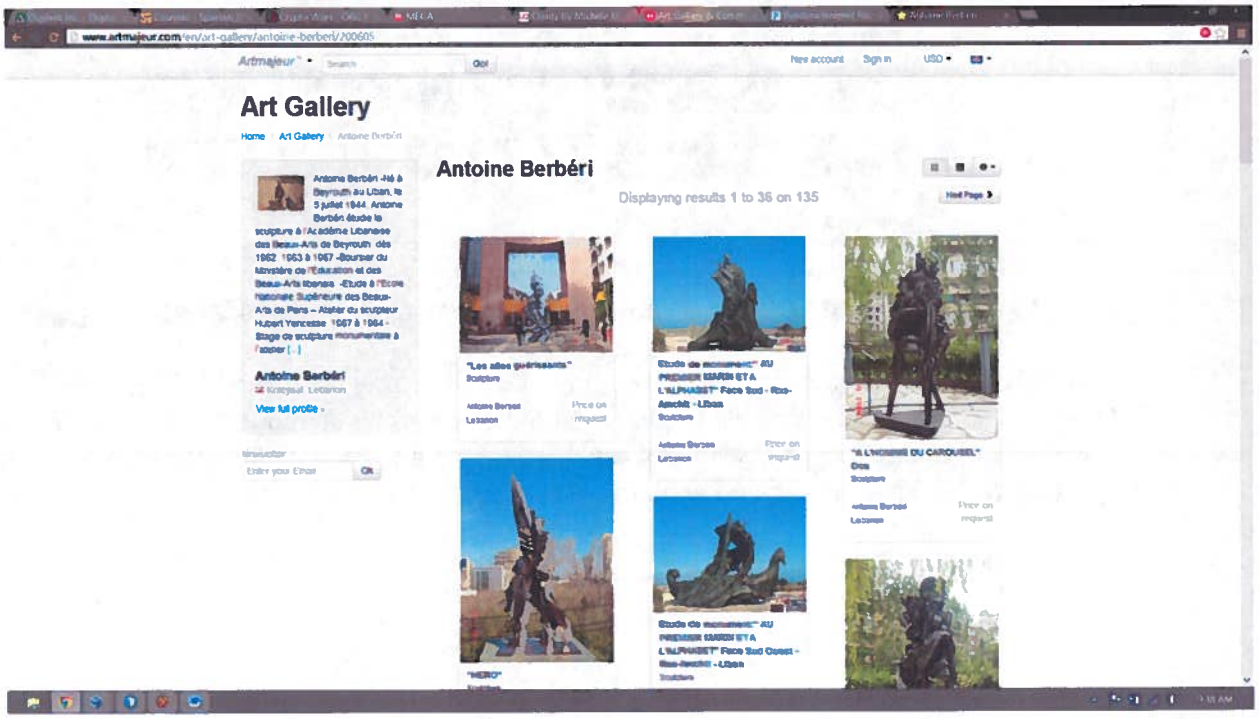


Figure 2.3a: This screenshot shows a gallery of artworks produced by a particular artist. This page can be accessed by clicking on any artist tags when viewing artworks.

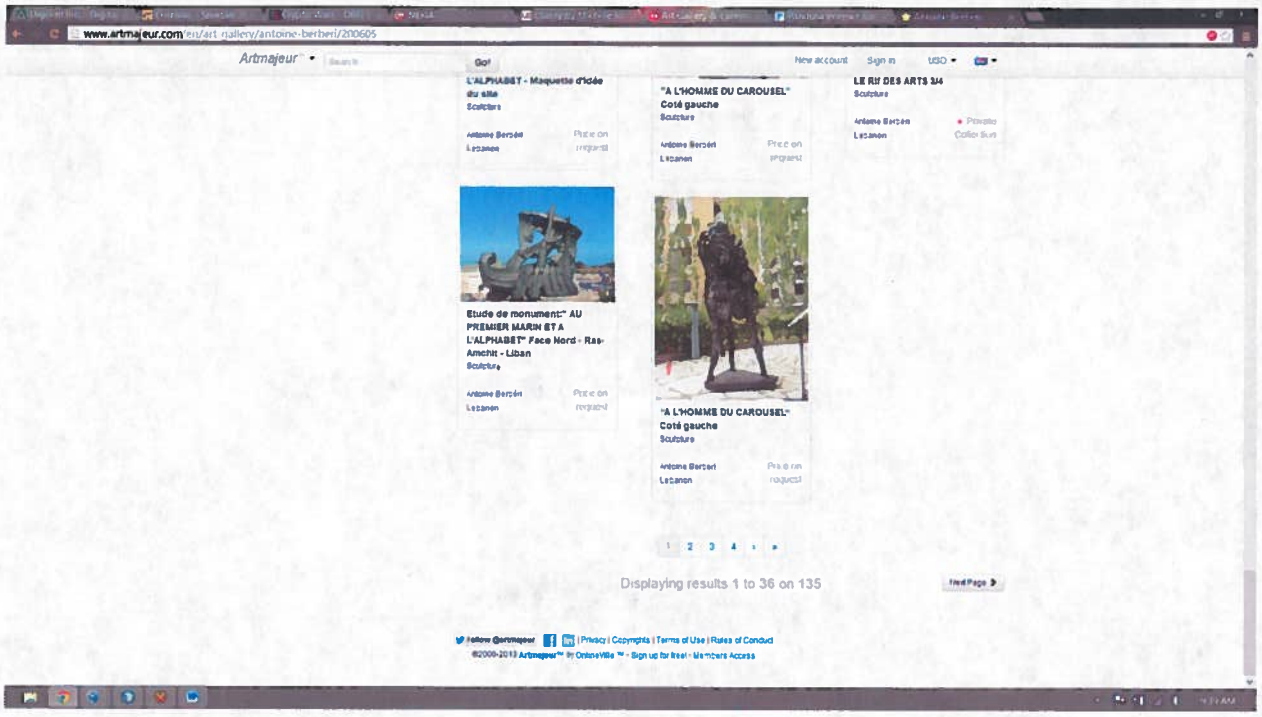


Figure 2.3b: The continuation of the Figure 2.3a.

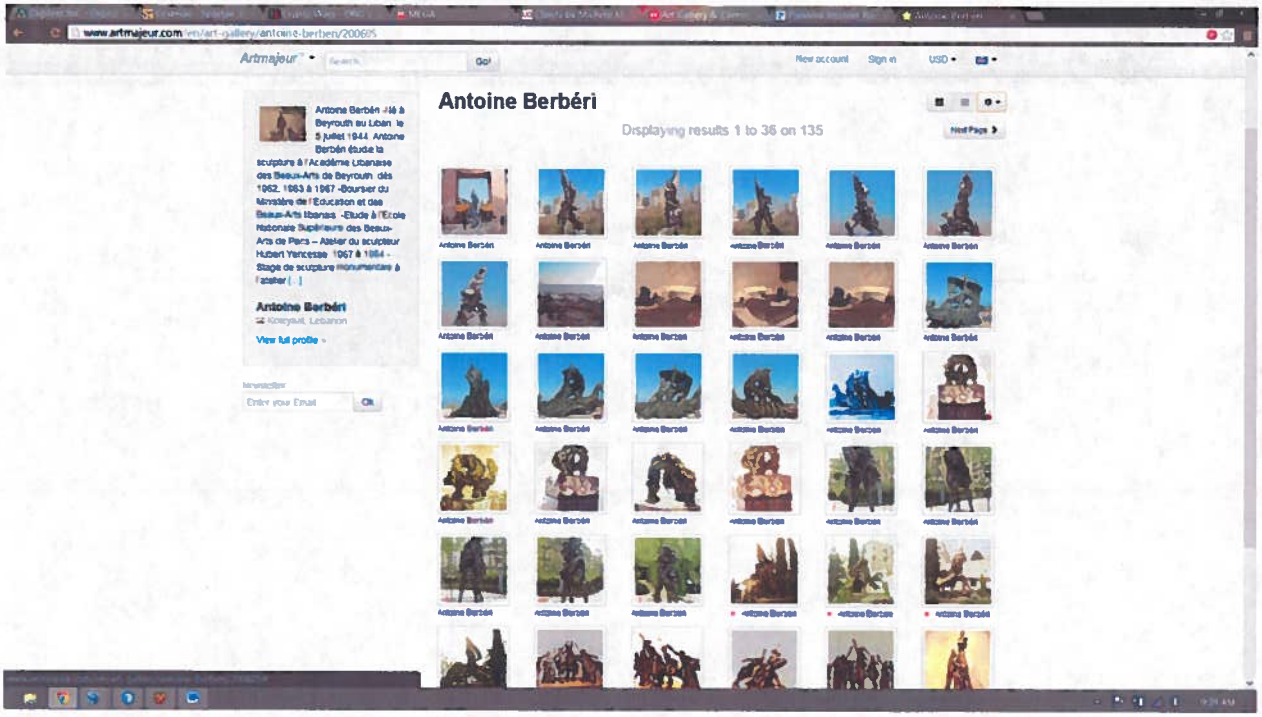


Figure 2.3c: Another gallery view of Figure 2.3a, this time being shown in tile format.

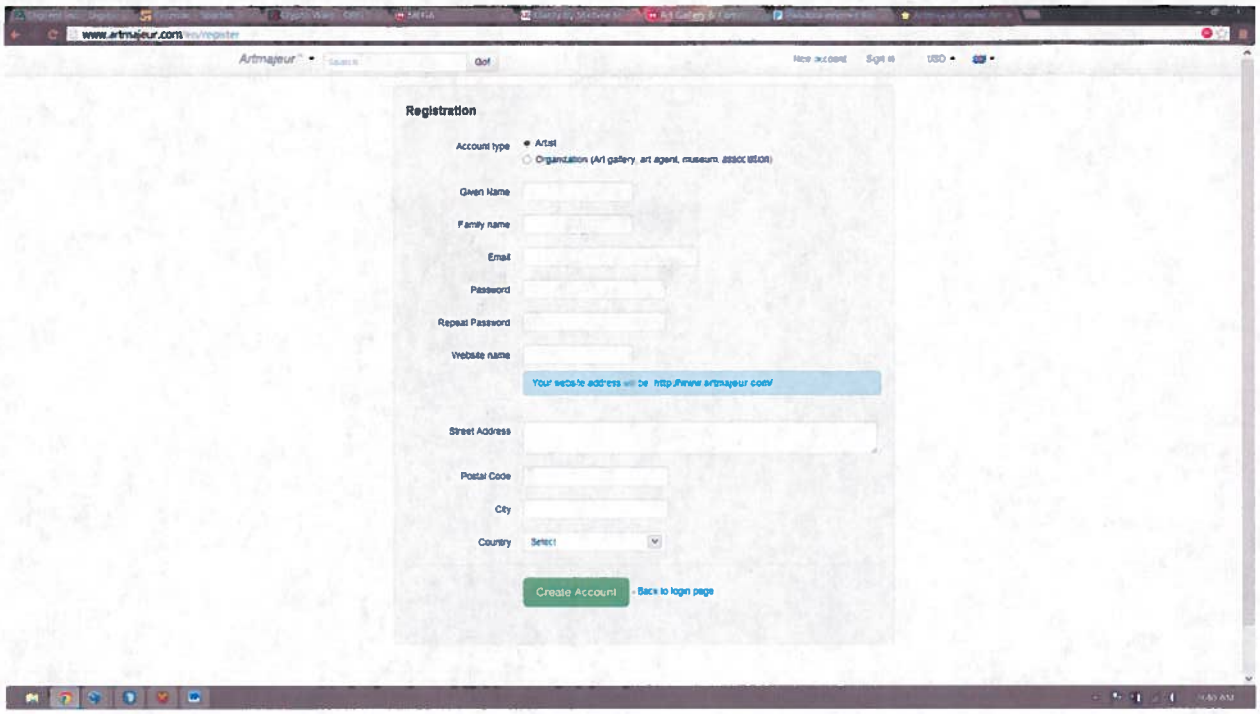


Figure 2.4: This screenshot shows the page used for new users wishing to register accounts with the website.

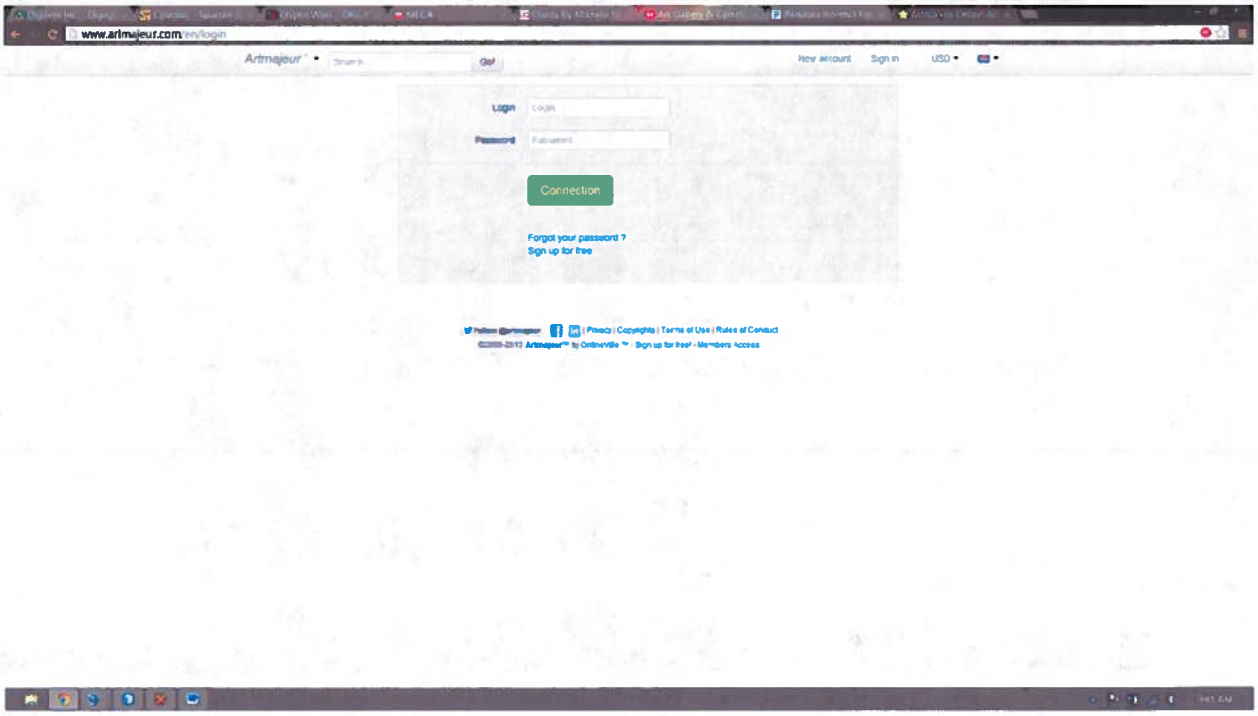


Figure 2.5: This screenshot shows the form for registered users wishing to log into their accounts.

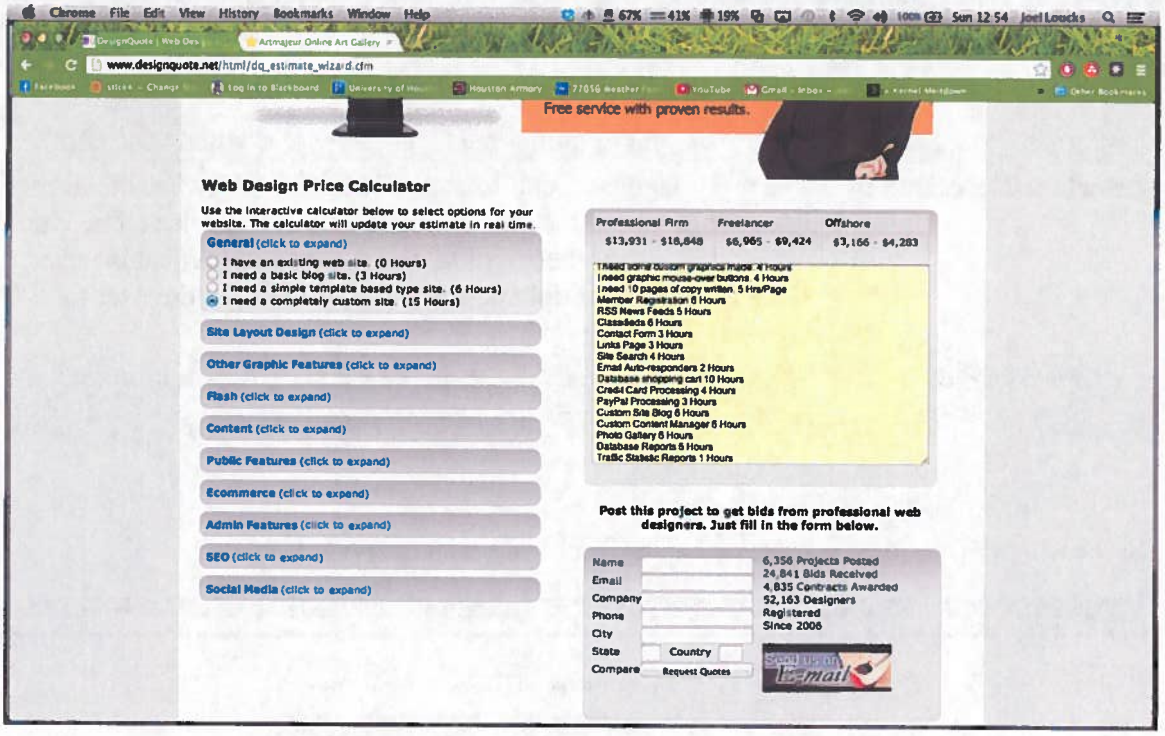


Figure 2.6: This screenshot lists the estimated costs for the development of the Artmajeur website given the features that were existent on the web pages.

Case Study 3	
Website Name	Red Bubble
URL	http://www.redbubble.com/
Technologies Used	Ruby on Rails running on a Cent/OS server with JavaScript.
Interface Description	The top image is a "call to action" which is showing the user a featured item that they are trying to push on the user. The email signup is also another "call to action." Pagination is also used since there are multiple pages of thumbnails for the user to browse.
Estimated Cost*	$3,000.00 for an offshore firm, $13,000.00+ for a professional firm

Screenshots of various web pages of RedBubble.com are provided below.

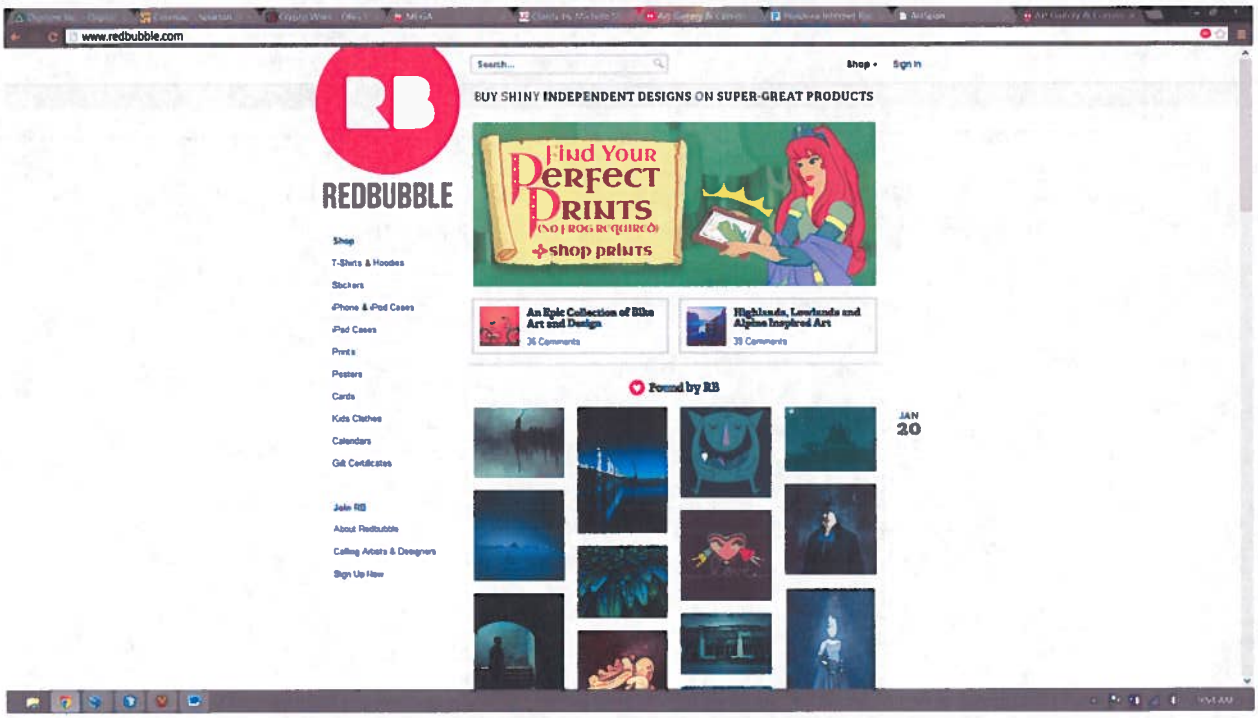


Figure 3.1: This is the screenshot for the home page of RedBubble. The user has the option of searching directly from the main page or signing into their account. Navigation is on the left hand side and will take you to the various pages associated with the website.

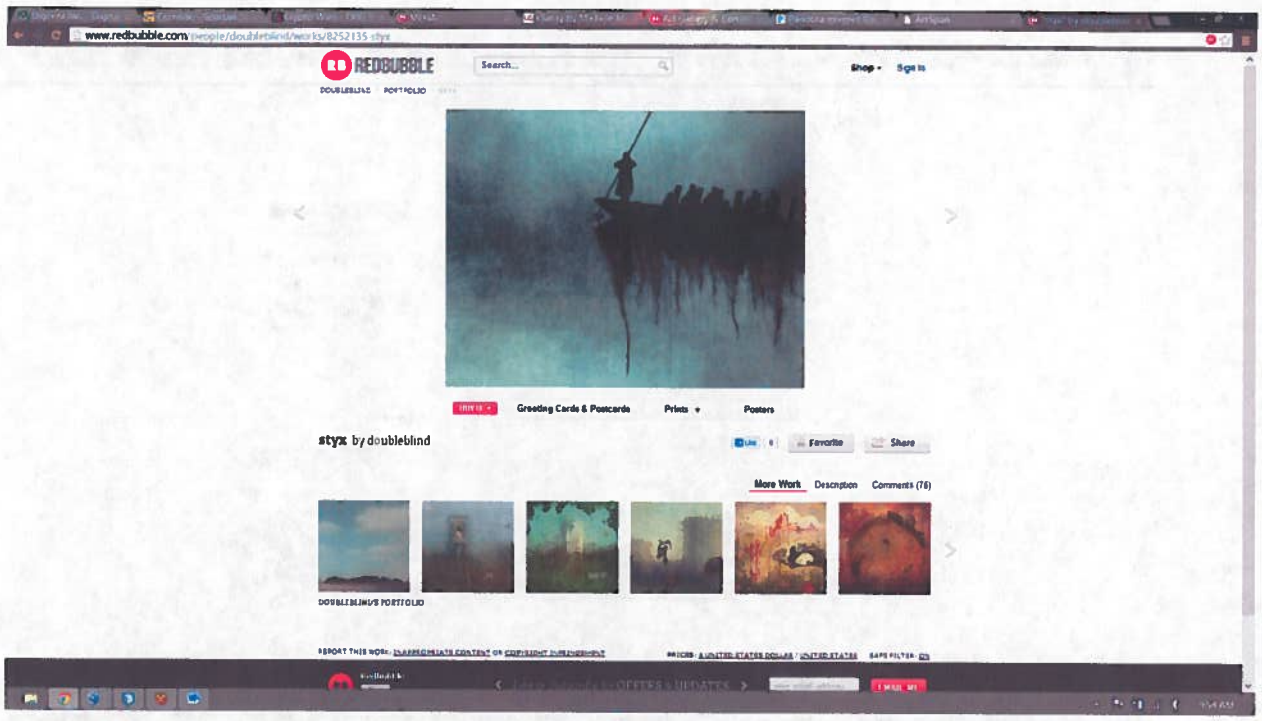


Figure 3.2a: This is a screenshot of a sample artwork. More artworks can be viewed by clicking on any image thumbnail in the section beneath the featured artwork. Also social network plugins exist where viewers can "like" the artwork and have it displayed in their Facebook feeds.

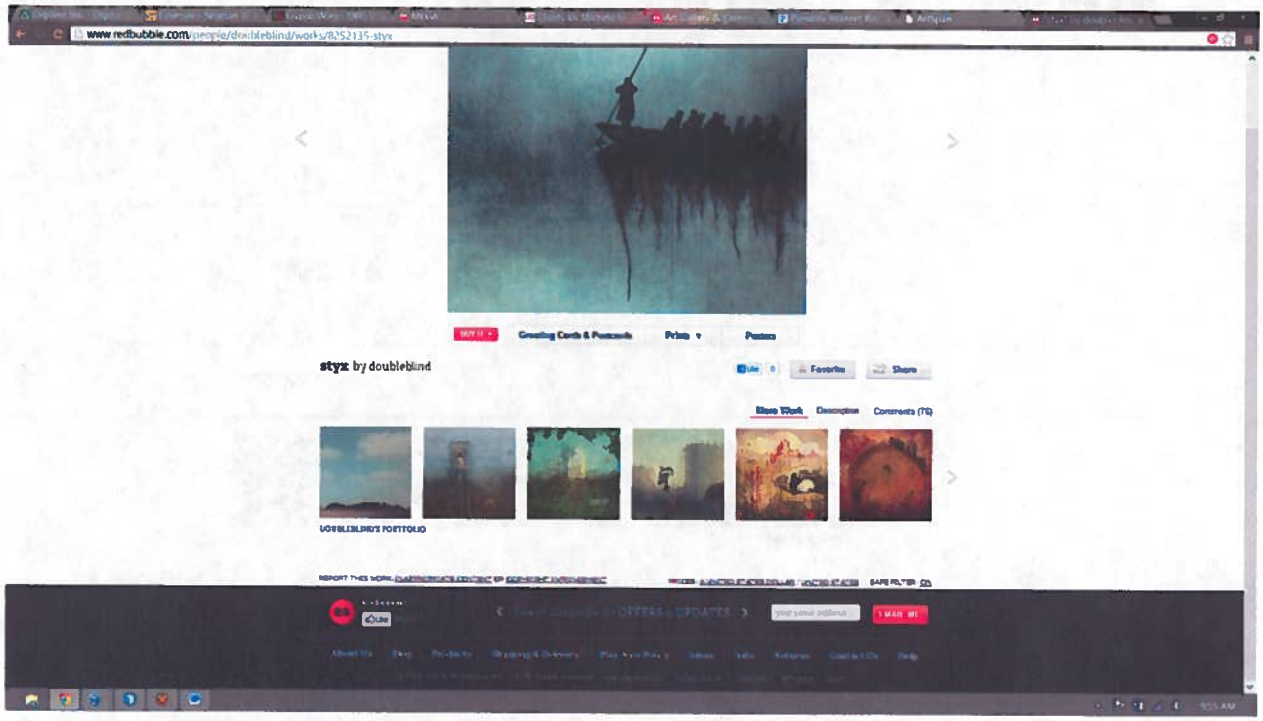


Figure 3.2b: This screenshot is a continuation of Figure 3.2a. The footer is shown.

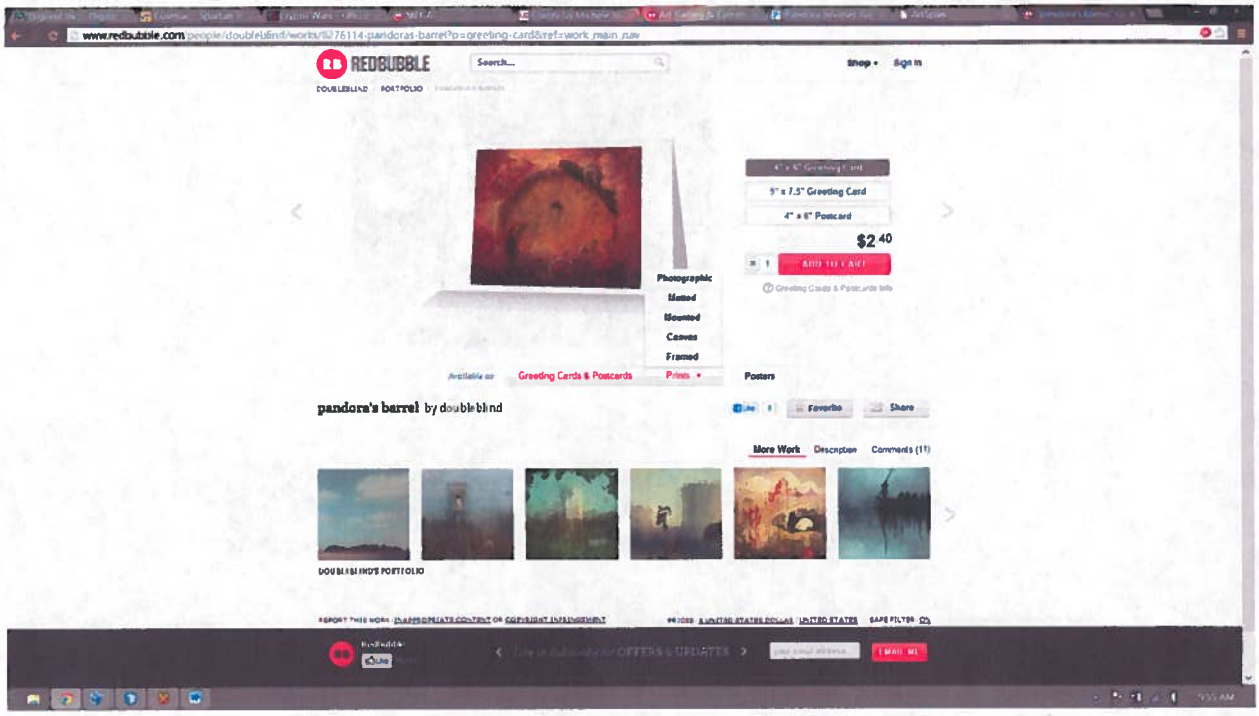


Figure 3.3: This is a screenshot showing a tab listing the different types of prints. It is currently showing the artwork for a greeting card print.

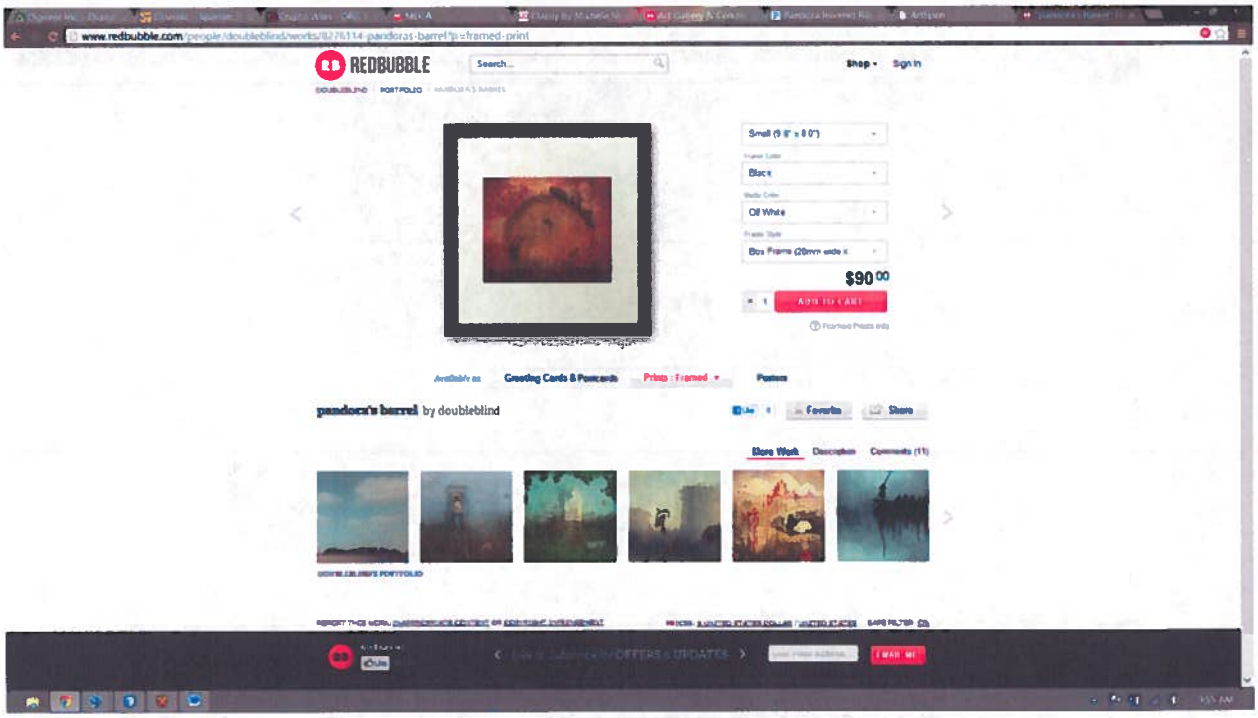


Figure 3.4: This is an example of an artwork with a virtualized frame. It allows the viewer to preview what their purchases would look like.

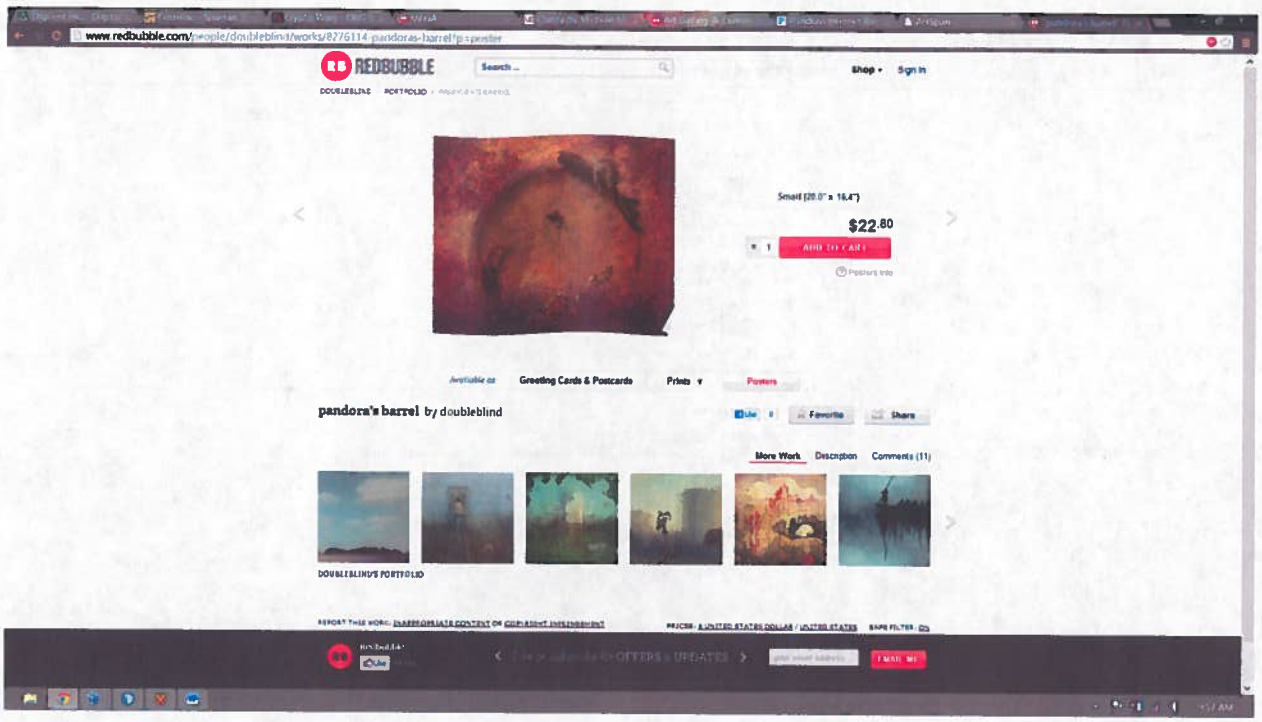


Figure 3.5: This is an artwork with a poster view.

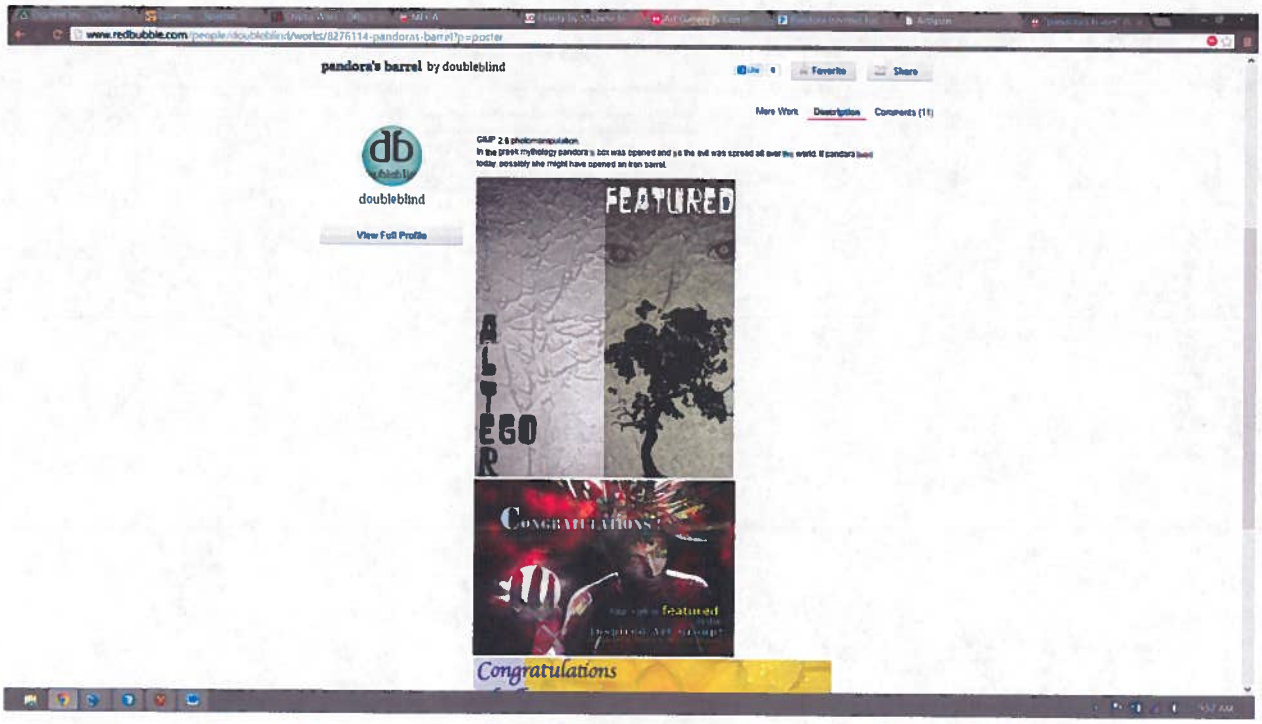


Figure 3.6a: A screenshot of another artwork but with the description view of the artist visible.

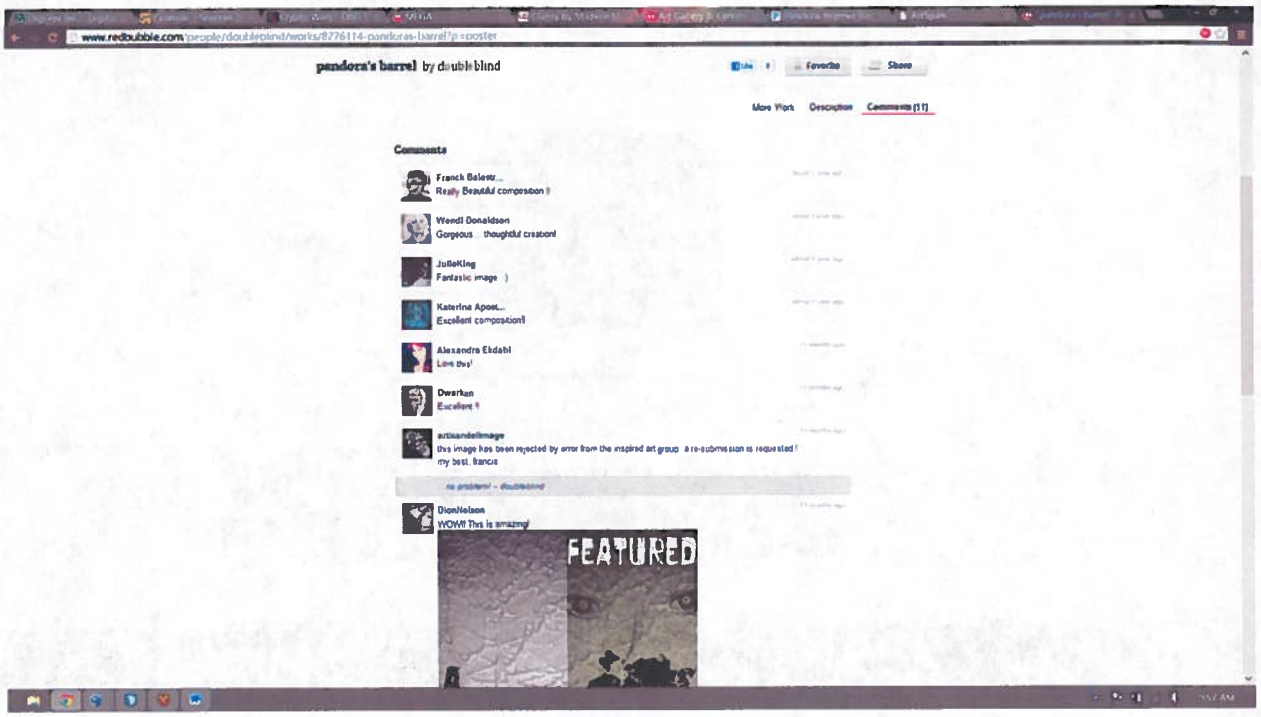


Figure 3.6b: The same page as Figure 3.6a but with the comments section visible.

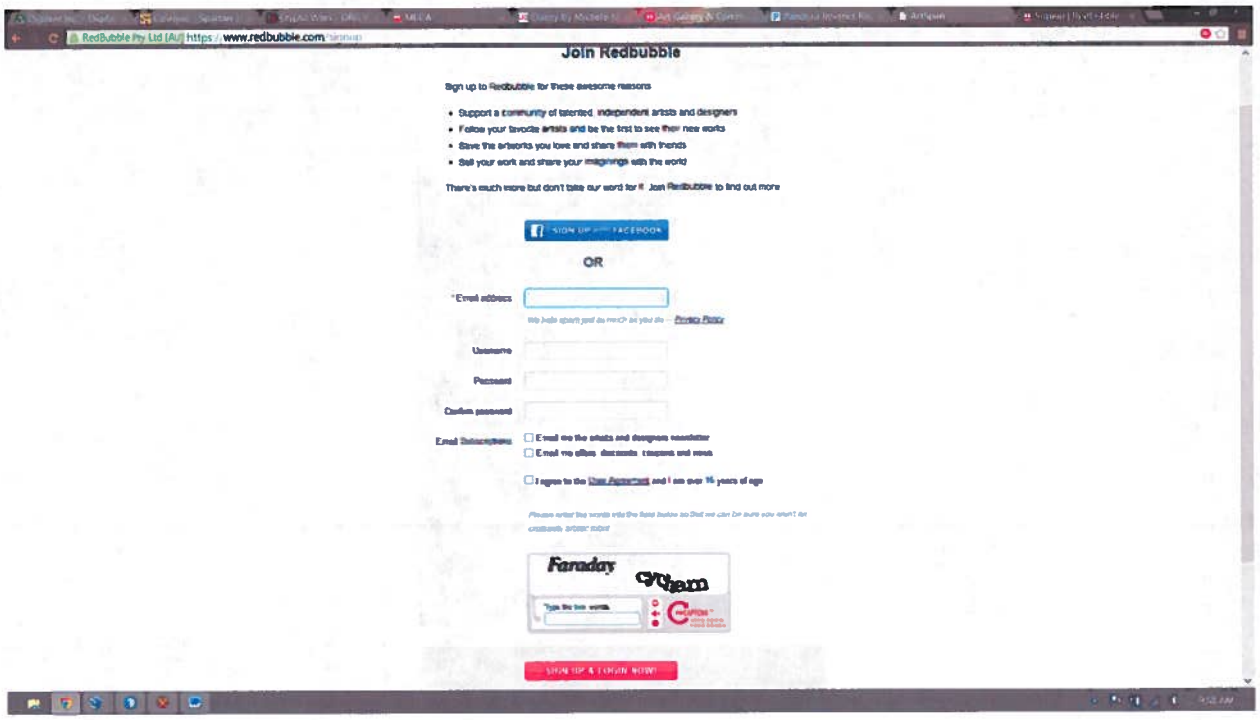


Figure 3.7: This is a screenshot of where new users can register accounts with the website. An alternative method of login exists where users can use their Facebook accounts as login credentials.

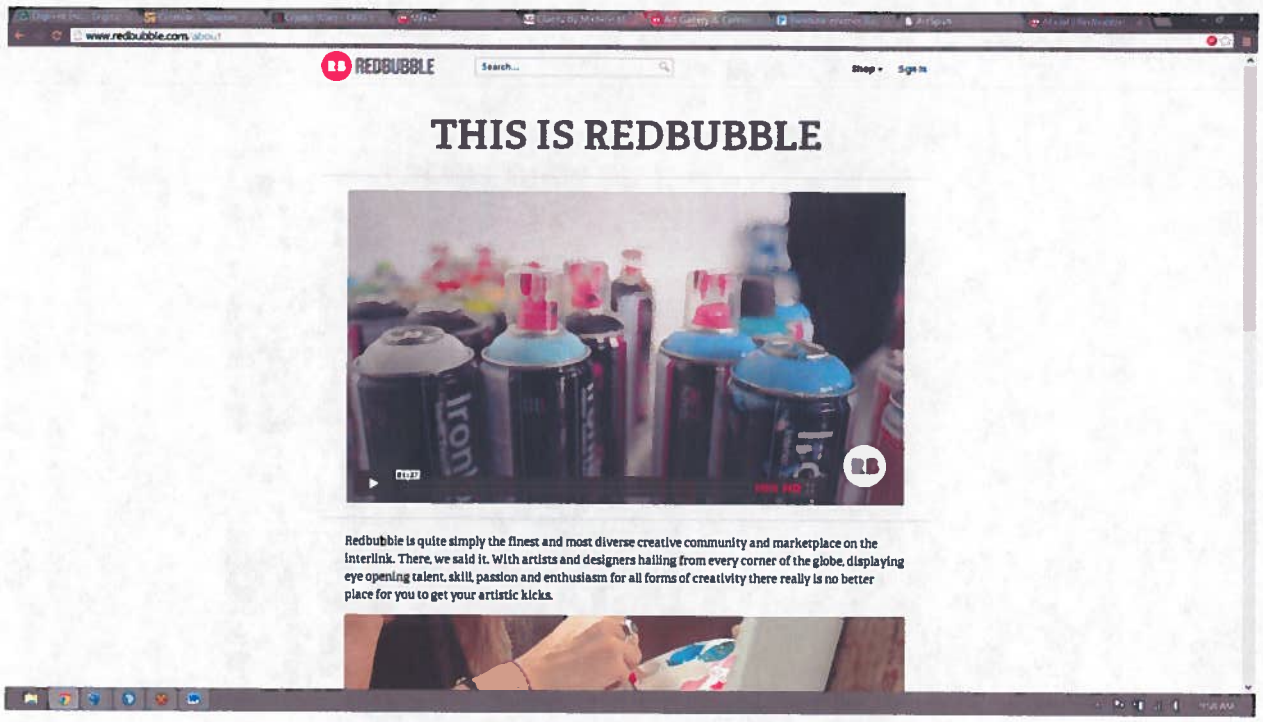


Figure 3.8a: This is the about page that explains who RedBubble is to the customer.

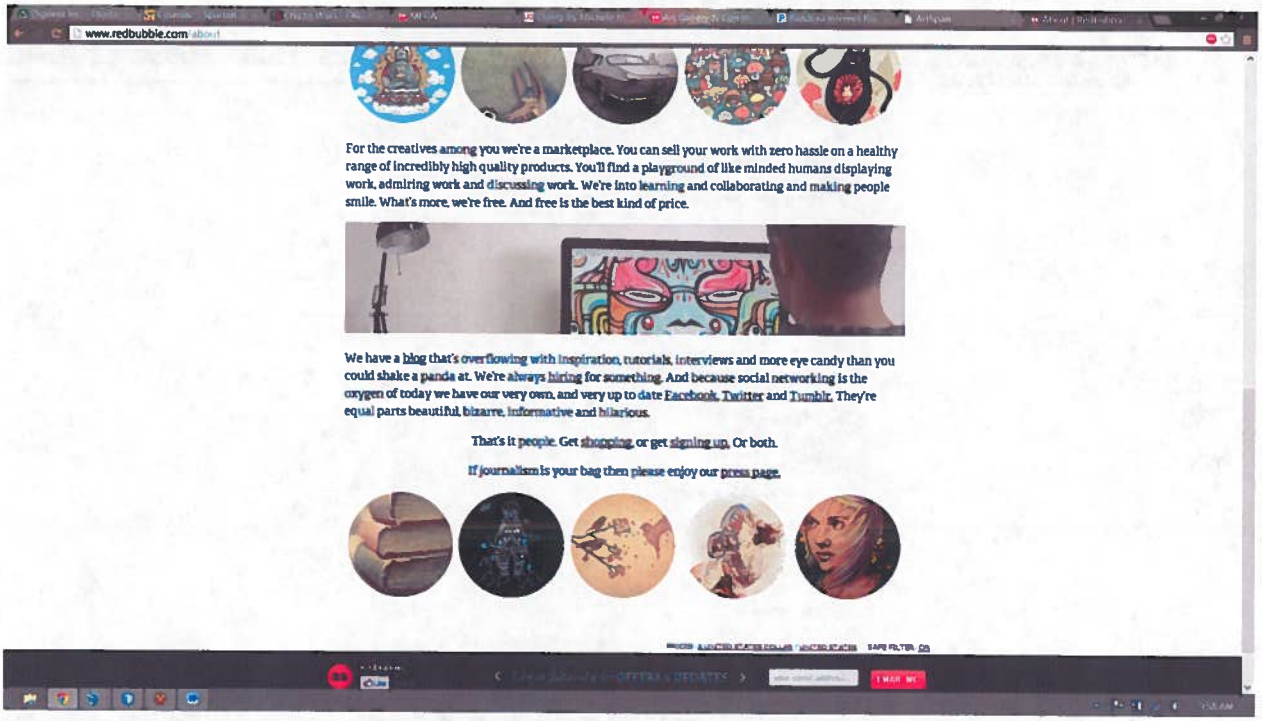


Figure 3.8b: Continuation of the about page.

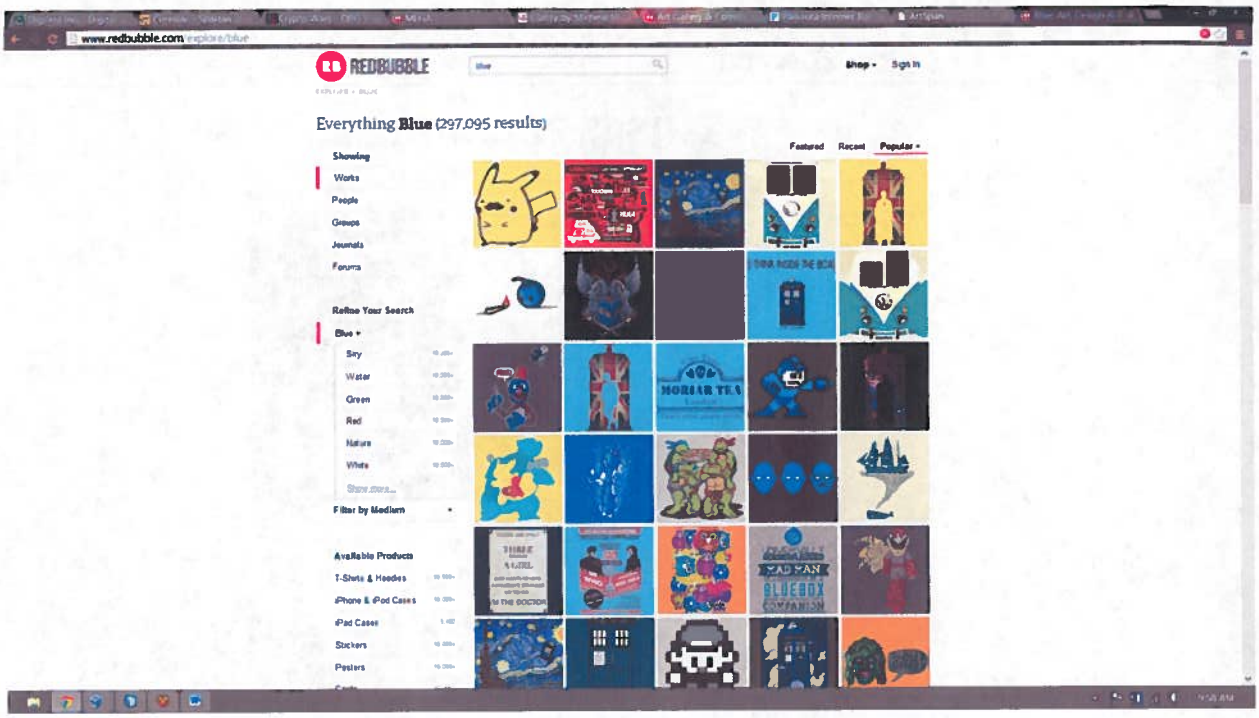


Figure 3.9: This is an example of a screenshot of returned search results after using the website search tool.

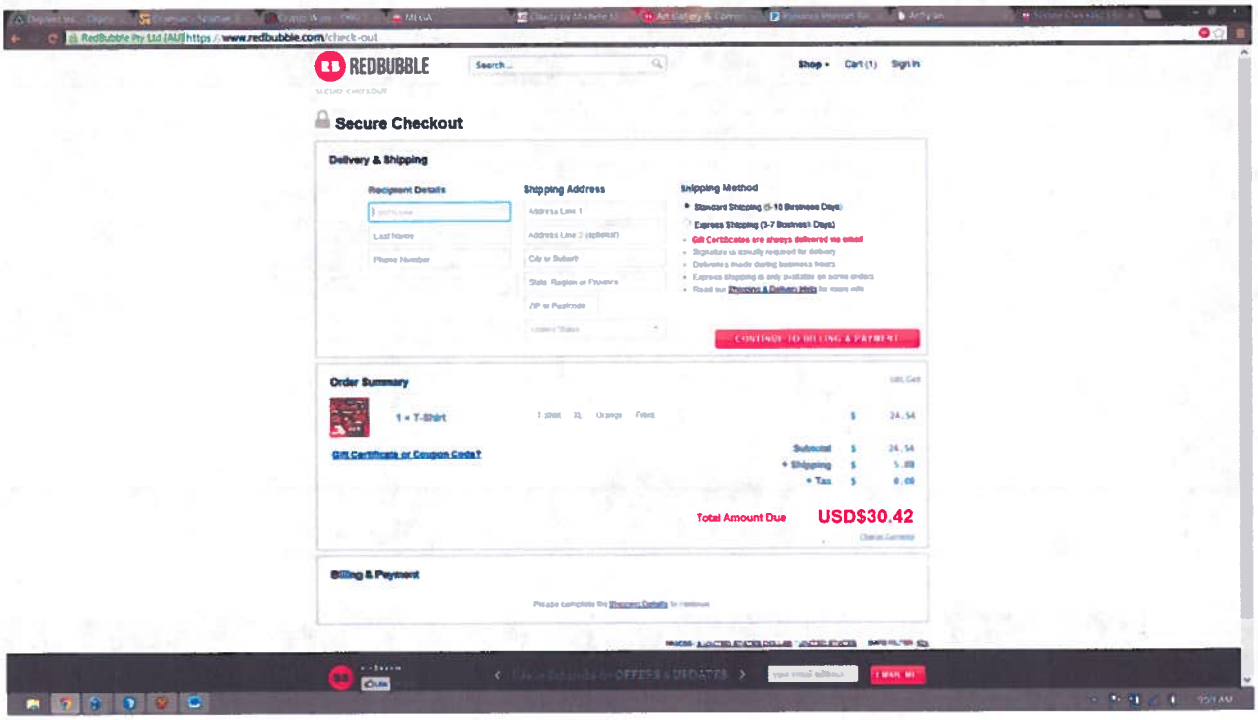


Figure 3.10: This screenshot shows the beginning of the checkout procedure for the customer.

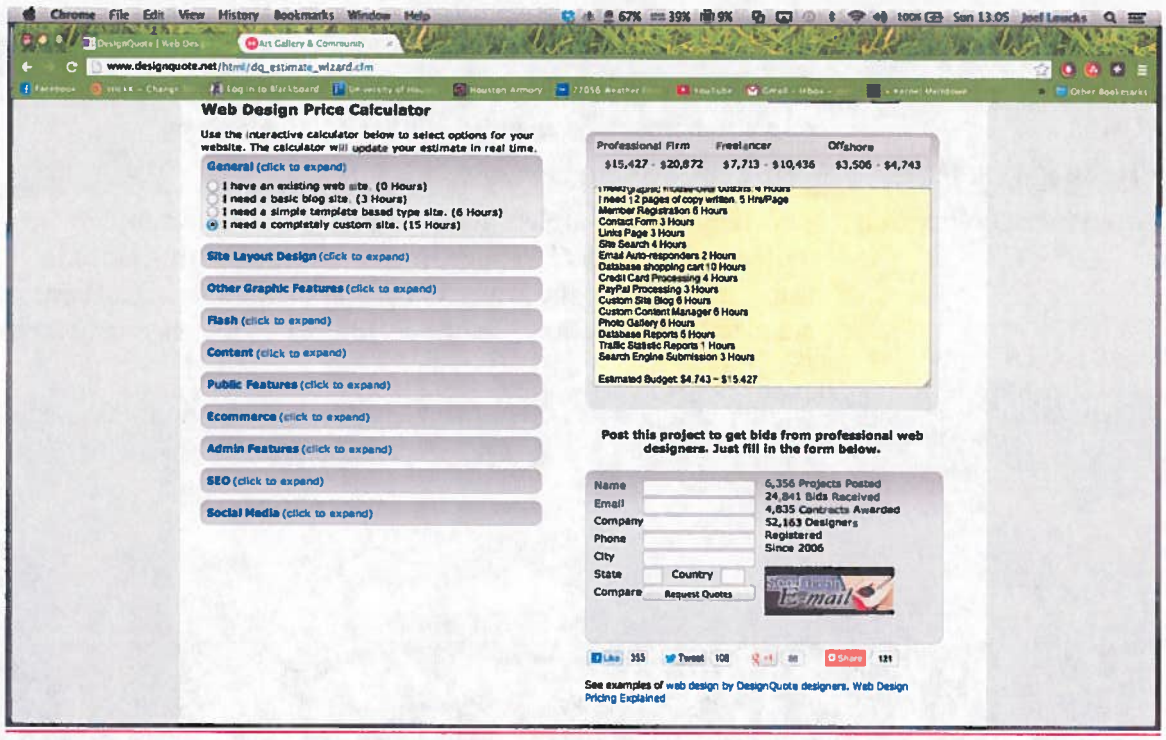


Figure 3.11: This screenshot lists the estimated costs for the development of the RedBubble website given the features that were existent on the web pages.

Case Study 3 Summary

RedBubble did some different implentations that really complement the site. The ability to view the artwork framed as well as the ability to view it printed on a T-Shirt give the organization a greater edge over their competition. RedBubble also has a cleaner and more streamlined interface when compared to other websites. It is not cluttered and has a fluid navigation for the user. One downside to the site is the lack of complete social interaction when compared to the other web pages. RedBubble only has Facebook whereas the other sites include Twitter and Pinterest.

Case Study 4	
Website Name	Sharon Norman Photography
URL	http://1-sharon-norman.artistwebsites.com/index.html
Technologies Used	PHP on an Apache server with JavaScript.
Interface Description	This page has a simpler layout on the front with a simple toolbar and header. The page implements pagination since it uses navigation to the artwork through the thumbnails. There are also "call to actions" in the search bar so the user can search for artwork.
Estimated Cost*	$3,000.00 for an offshore firm, $13,000.00+ for a professional firm

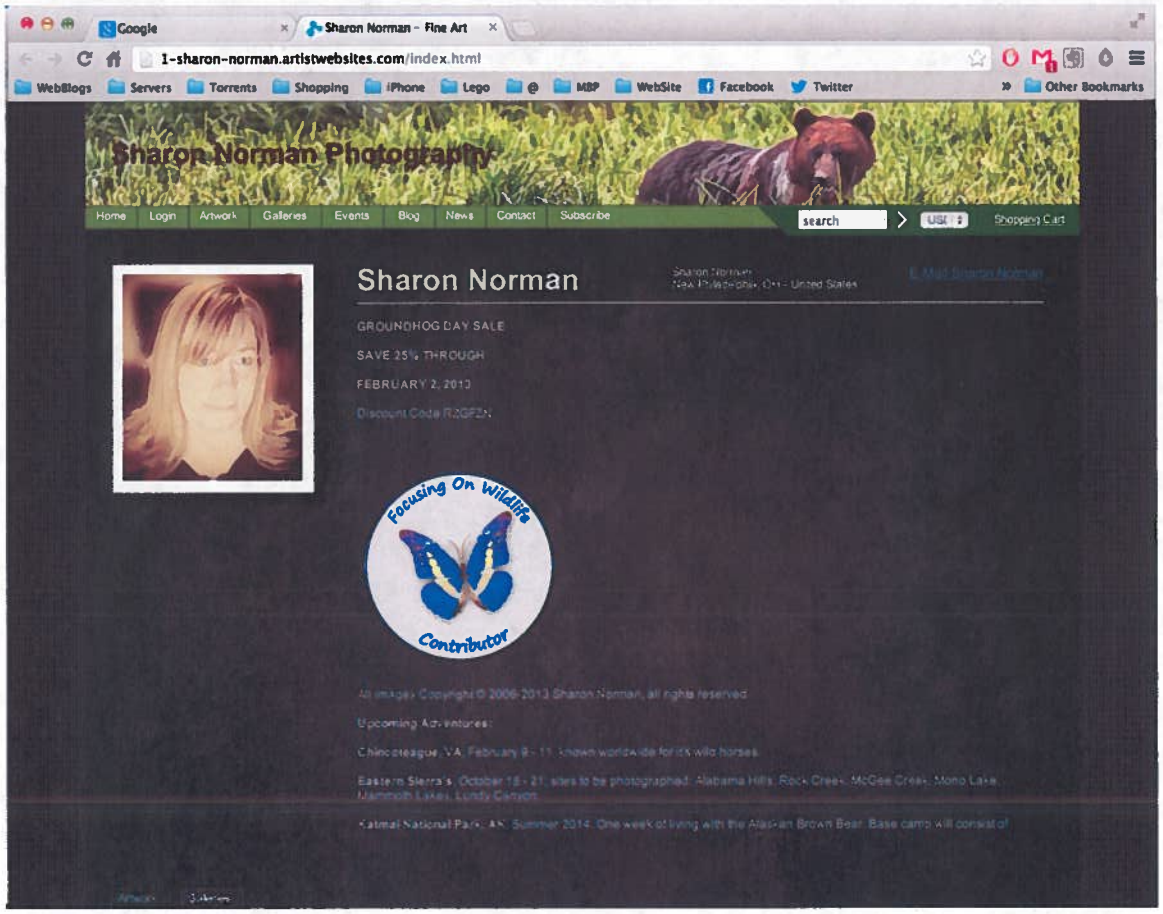


Figure 4.1a: This is the home page for the Sharon Norman Photography website. It has a basic layout and some brief information about the company.

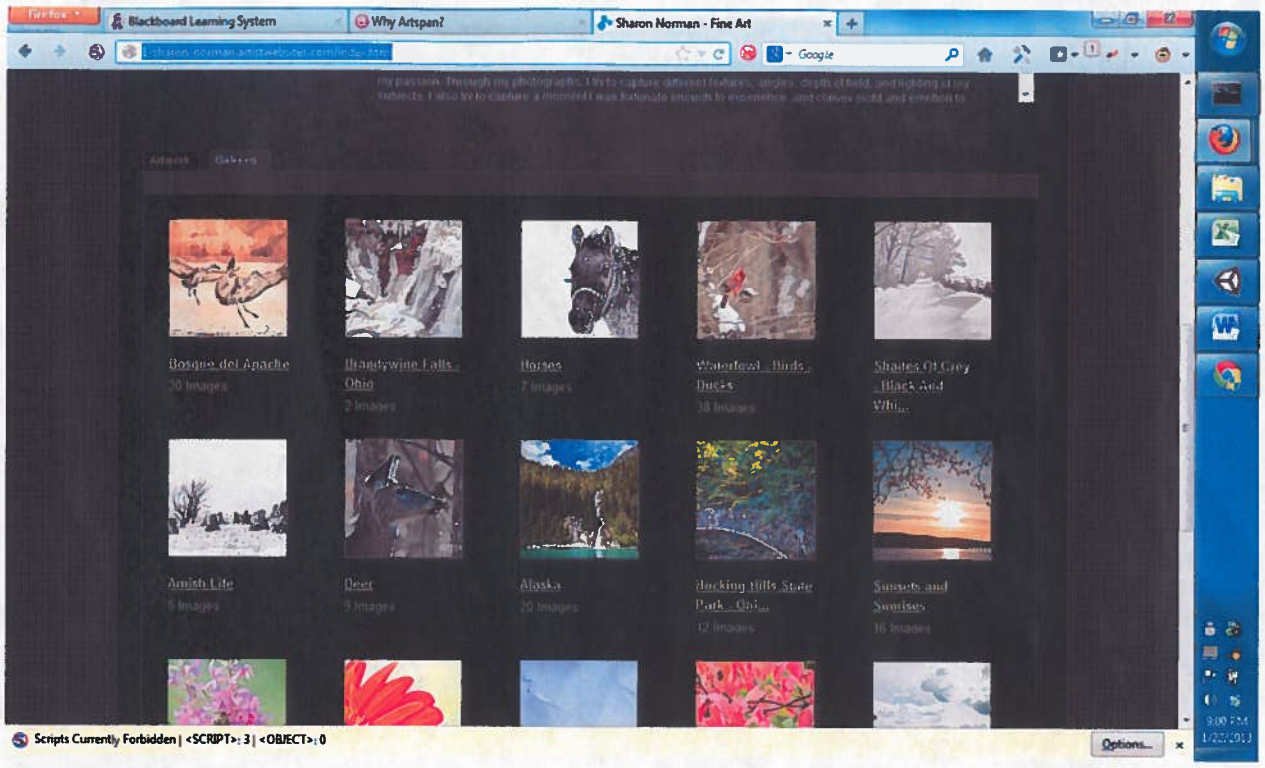


Figure 4.1b: Continuation of the home page. A gallery listing artwork images is shown below the main content on the home page.

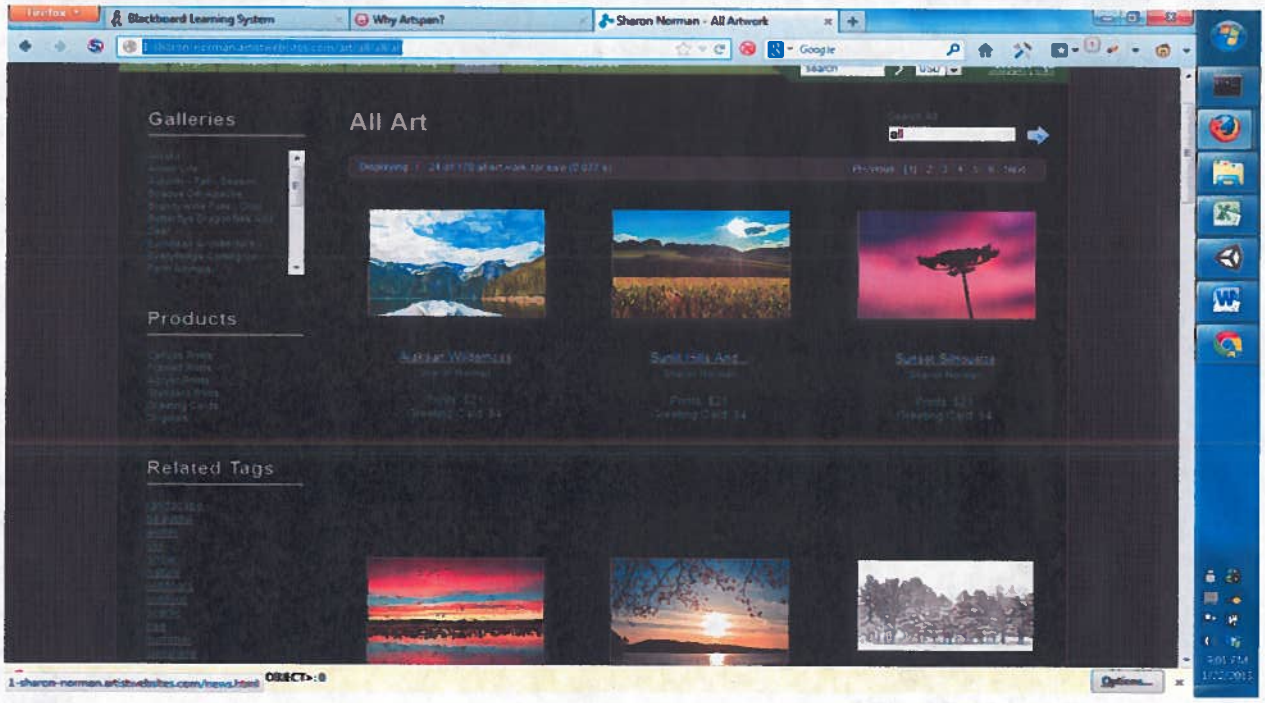


Figure 4.2: In this view, all artwork images are listed. The viewer also has the ability to navigate to different artworks using the navigation menu to the left.

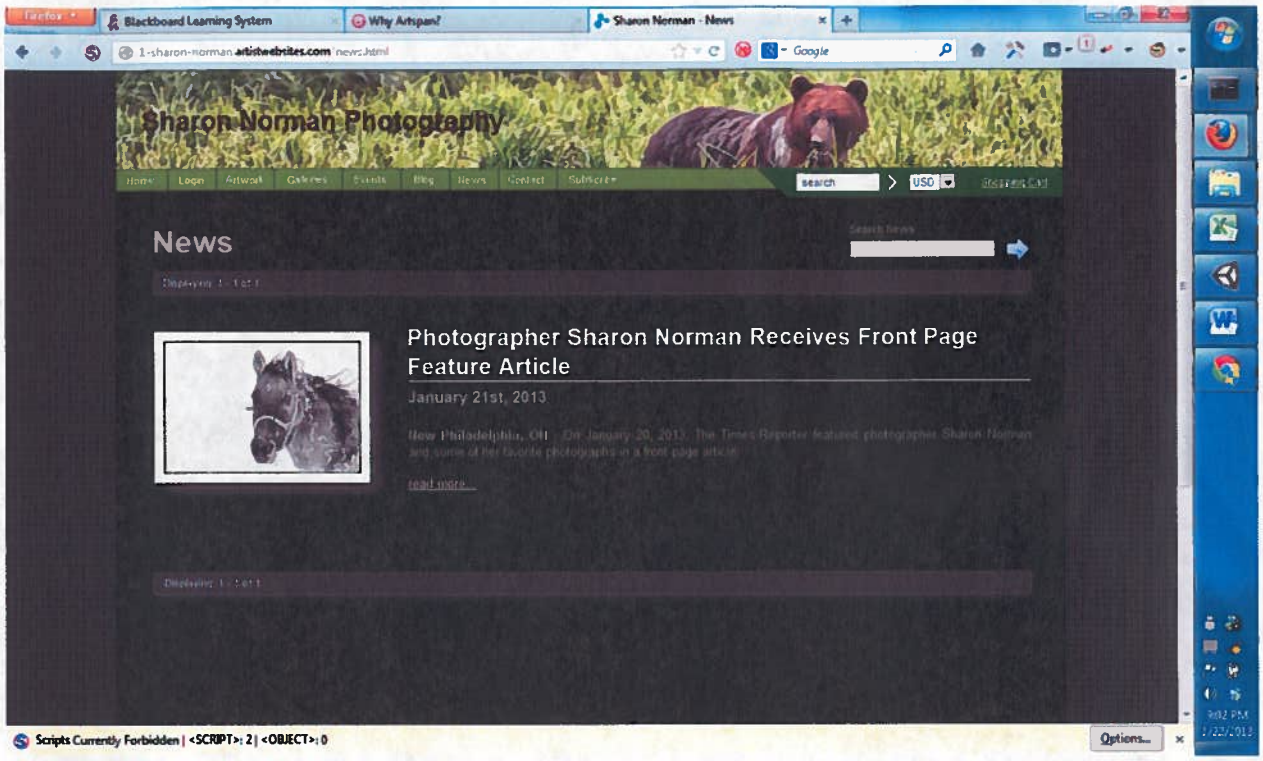


Figure 4.3: This is a screenshot of the news articles page.

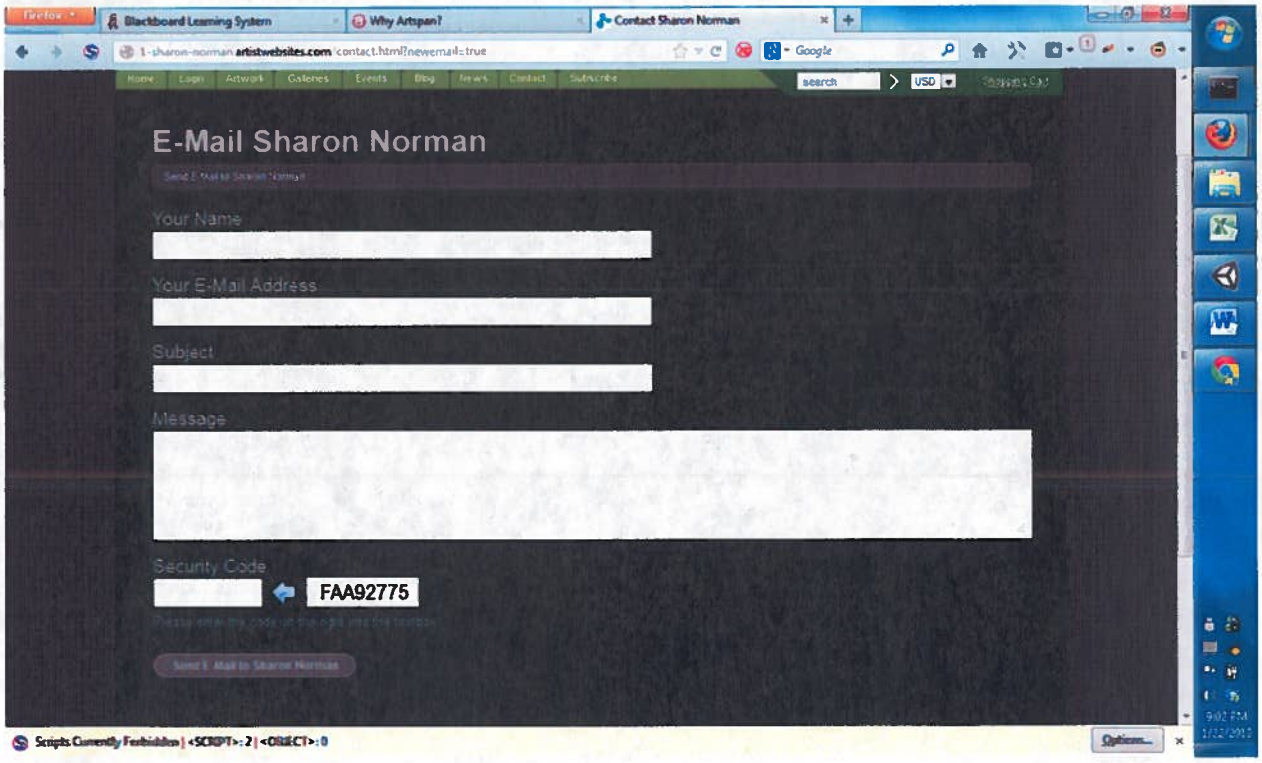


Figure 4.4: This screenshot shows a contacts form that a viewer can fill out to send an email to the site owner.

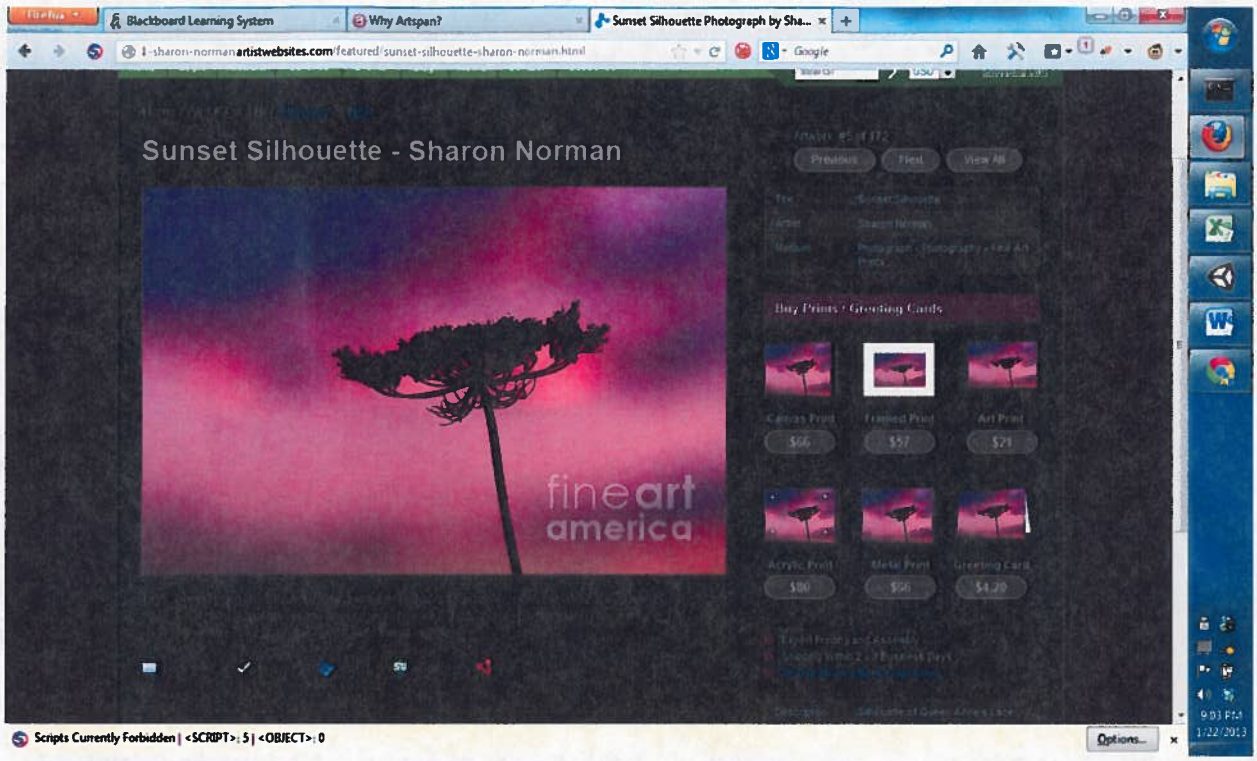


Figure 4.5: Screenshot featuring an artwork on sale. Notice that the title, artist, and medium are displayed in a panel to the right of the featured artwork.

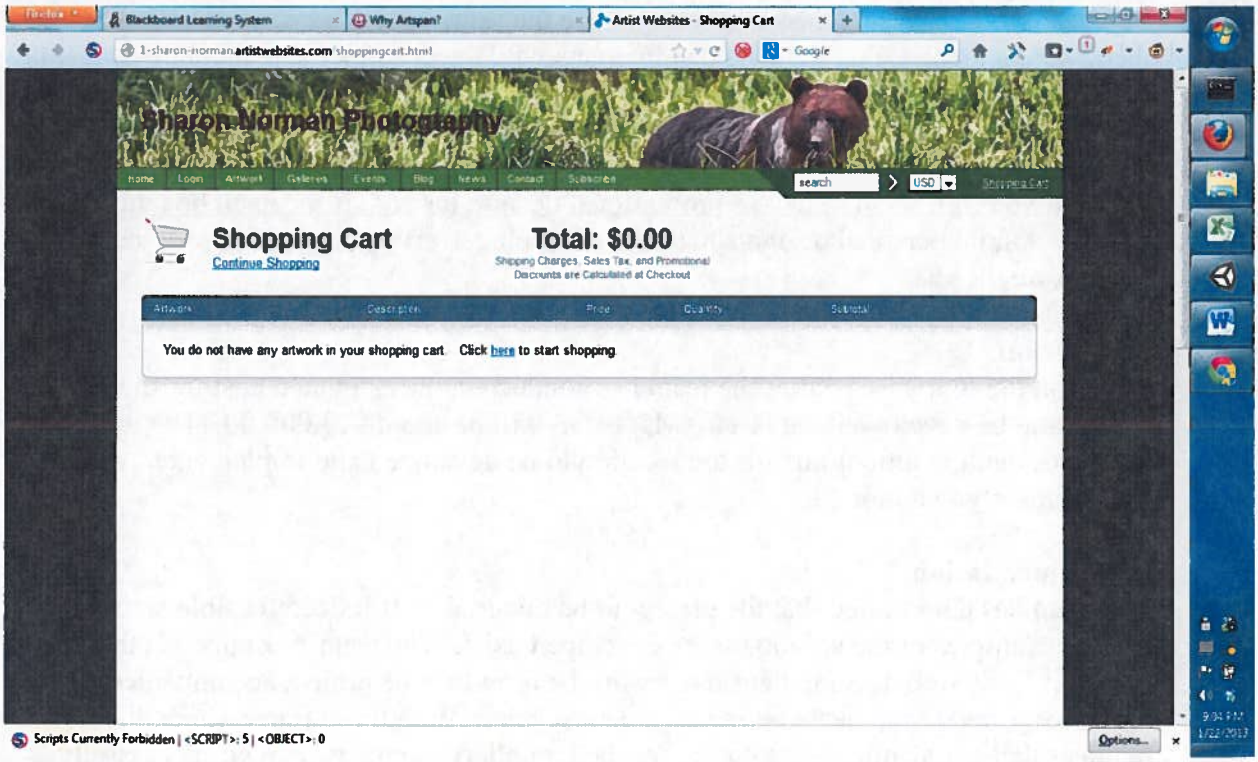


Figure 4.6: Screenshot of the checkout cart.

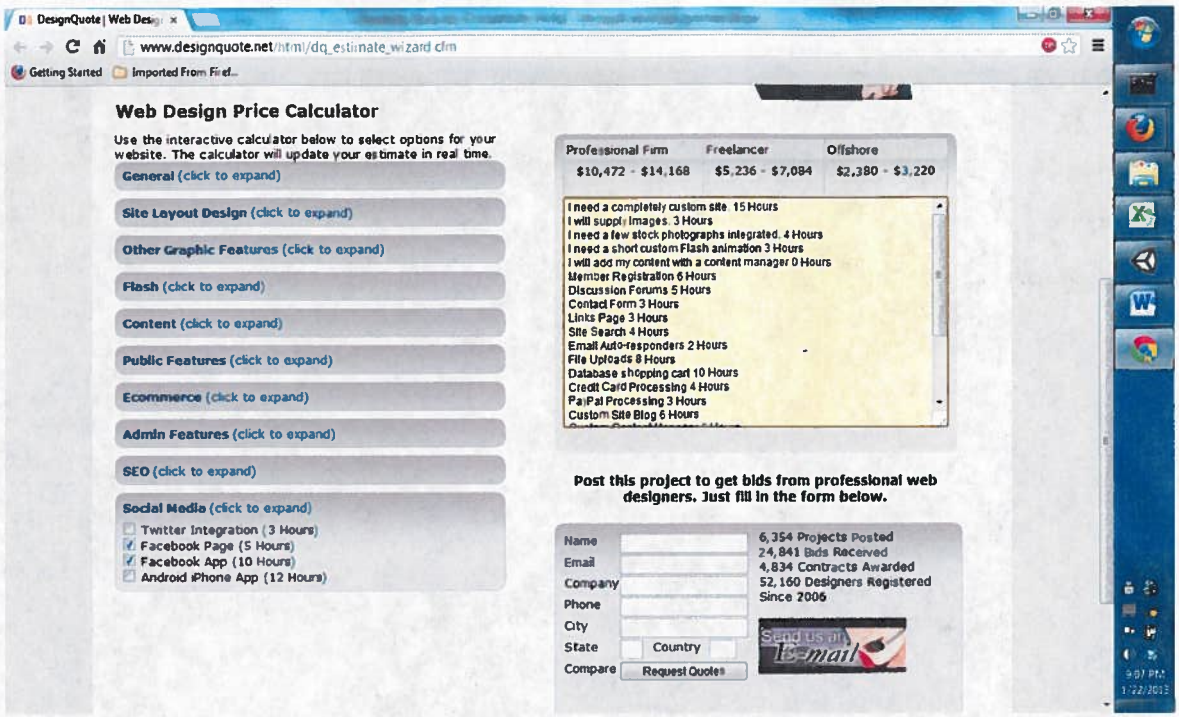


Figure 4.7: This screenshot lists the estimated costs for the development of the Sharon Norman Photography website given the features that were existent on the web pages.

Case Study 4 Summary

The four websites researched essentially share the same goal. The companies behind them want a presence on the web that interacts with the customer and encourages them to make their purchase with them instead of a competitor. Of the four websites shown, the fourth, Sharon Norman Photograpy, stands out from the rest in terms of informal design. By looking at the site it is evident that the quality is not at par with that of the other three in that it does not seem to have a professional design. Instead, it seems to be tailored towards a more personal design since the artist being avertised is the only one selling artwork on the site.

Conclusion

Based on the four case studies the team has conducted, the estimated cost for the website solution to be developed that is TEAM5OAGS will be about $13,000.00. The team also estimates that the time frame for the solution to be developed and implemented will be approximately two months.

Recommendation

The team has determined that the project to be undertaken is indeed feasible since many similar examples of the solution to be developed exist. The team recommends that the Ruby on Rails web development framework be used for this project accompanied with the MySQL relational database management system. With these frameworks, the team believes that a dynamic web solution for the art gallery company can be successfully developed and implemented within the time, budget, and quality constraints.

* Estimated costs for the development of websites were obtained through http://www.designquote.net/html/dq_estimate_wizard.cfm.

RED TEAMS DELIVERABLES FEEDBACK "blue print" UML Modeling with actual examples.

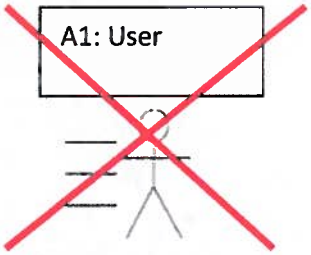
MODELING Definitions:

- "The Requirements" - "are contained on pages 1, 2, ... and 11 starting with Line Number 1 and ending with Line Number 147". Please think of it as a .pdf file that cannot be modified.
- "The Requirements" – are not repeated for Step 1: Identify Actor, then copied again, thus here there are again on pages 12, 13, 14, and 23 for Step 2: Identify Use Cases; they stay on pages 1, 2, ... and 11 with Line Numbers 1 to 147 only. You apply Step 1 on "The Requirements", and then you apply Step 2 on "The Requirements". Think of "The Requirements" as being a .pdf 11 pages document. I can scan it if you desire so. If you Requirements have less than or more than "The Requirements" you lost up to 10 points.

1. Feedback "SE Team Project with Line Numbers for UML USE CASE DIAGRAM.doc"

USE CASE MODELING:

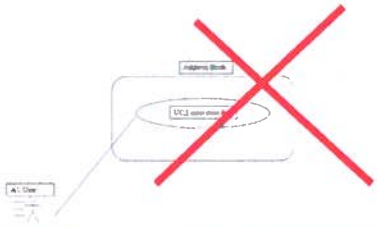
- Step 1 template does not have a box around the A Label like this:

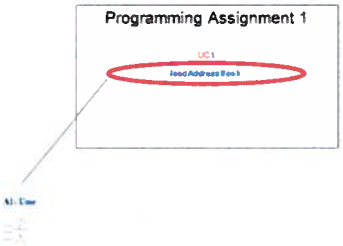


- Step 2 template does not have a box around the UC Label like this:

If you did not use the UML USE CASE DIAGRAM templates you lost up to 20 points.

- Step 3 is the "Select All Templates you placed/annotated on "The Requirements" i.e., pages 1, 2, ... and 11 ", "Cut" , and "Paste" them on a fresh page 12, then rearrange according to the UML USE CASE DIAGRAM notation:





NOTE: There is no need that on page 12 you present a summary of the UC s the only place they exist is as annotations you placed on "The Requirements" i.e., pages 1, 2, ... and 11 as part of Steps 1 & 2. What goes on page 12 is the UML USE CASE DIAGRAM in one place not a different one per each actor where UCs are repeated.

NOTE: UML USE CASE DIAGRAM must be created in Word. If you did not use Word you lost up to 20 points.

If the UML USE CASE DIAGRAM does not match 100% the annotations you placed on the "The Requirements" i.e., pages 1, 2, ... and 11 you lost up to 30 points.

- Step 4 is the "Scenario" Description of each Labeled Use Case:

~~UC:1 This prompts the use to enter that data and then stores it in the appropriate places.~~

Must use a Template (see MRT 1 Tutorial for one or just something like this):

UC1: showSelection	
Actors:	A1: Customer
Brief Description:	To allow Customer to interact with the Candy Machine
Overview:	Menu with Welcome to Shelly's Candy Shop To select an item, enter: 1. for Candy 2. for Chips 3. for Gum 4. for Cookies 9. to exit

NOTE: The UC s become methods in the UserController Class of the MVC Design Pattern. In this class you are required to use MVC. Thus, use the "C" compilable module convention.

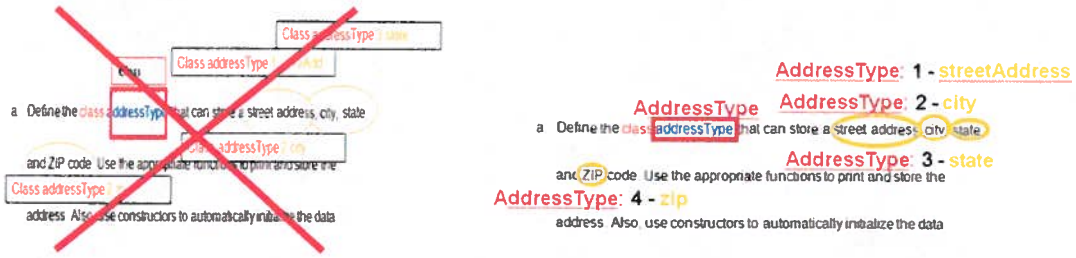
~~enter/store data~~ searchByLastName

NOTE: Unless your UML Use Case Model is UML "compilable" you will be considered as having no "**blue print**" for the purpose of building the "**house**". If you did not use the UML USE CASE Description templates you lost up to 30 points.

2. Feedback "SE Team Project with Line Numbers for UML CLASS DIAGRAM.doc"

USE CLASS MODELING:

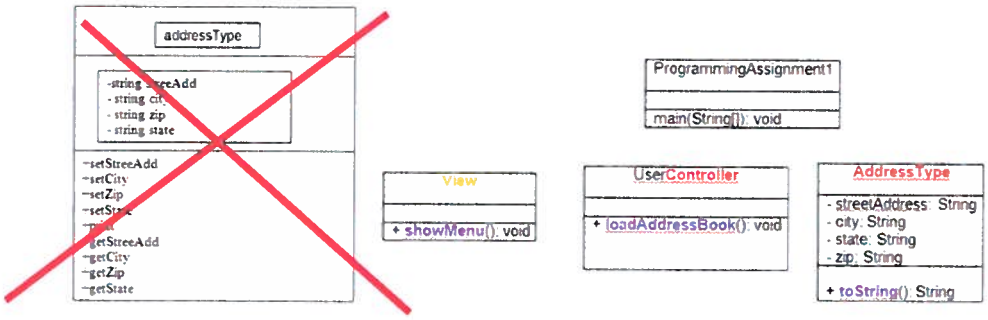
- **Steps 1, 2** templates do not have a box around the **Class** or **Attribute** **Labels** like this:



NOTE: Classes start with UpperCase Letter by Convention.
NOTE: Attributes start with lowerCase letter by Convention.

If you did not use the UML CLASS DIAGRAM templates you lost up to 50 points.

- **Step 4** is the "Select All Templates you placed on "The Requirements" , "Cut" , and "Paste" them on a fresh page 12, then rearrange according to the UML CLASS DIAGRAM notation. If the UML CLASS DIAGRAM does not match 100% the annotations you placed on the "The Requirements" you lost up to 40 points. If the UML CLASS DIAGRAM is not MVC you lost up to 20 points.



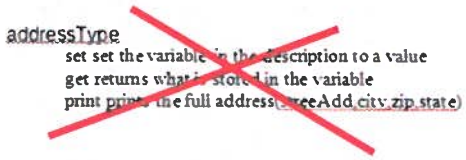
NOTE: There is no need that on page 12 you present a summary of the Classes, Attributes. The only place they exist is as annotations you placed on "The Requirements" as part of Steps 1, 2. What goes on page 12 is the UML CLASS DIAGRAM.

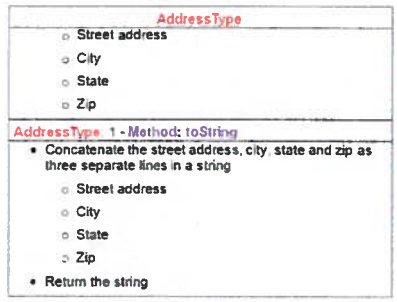
NOTE: UML CLASS DIAGRAM must be MVC, i.e., you need to add the **UserController** Class (UML USE CASE DIAGRAM, "right click" or "copy and paste" 100% match) and the View Classes.

NOTE: UML CLASS DIAGRAM must be created in Word.
If the UML CLASS DIAGRAM is not created using Word you lost up to 20 points.

This is Feedback for when you will do OOD:

- **Step 5** is the "Pseudocode" is for methods not for classes. Methods must have parameters and return types. Think of "Right Click" on the UML Class DIAGRAM "blue print" to "auto" generate the "house" skeleton classes needed for TDD.





NOTE: Think of "Right Click" on the UML Class DIAGRAM "blue print" to "auto" generate the "house" skeleton classes needed for TDD.

This is Feedback for when you will do TDD:

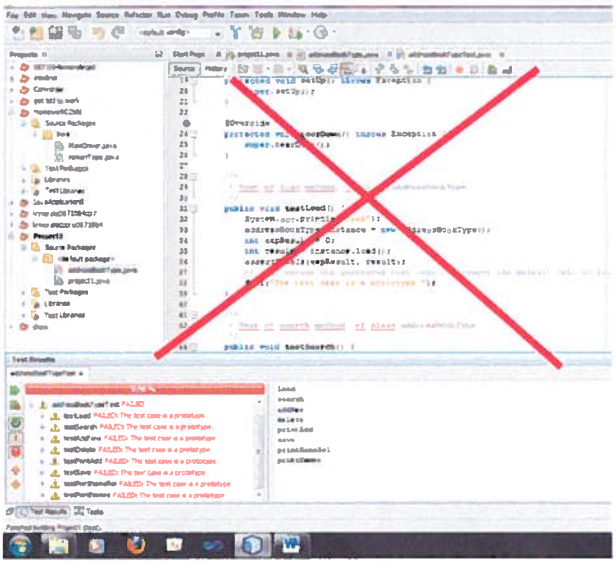
USE TDD Testing Definitions:

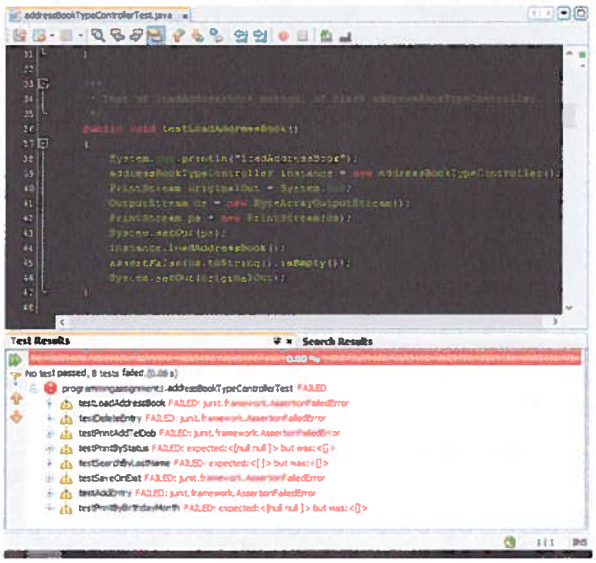
NOTE: Think of "Right Click" on the UML Class DIAGRAM "blue print" to "auto" generate the "house" skeleton classes needed for TDD in JAVA/C#/PYTHON/ RUBY/PHP.

- **Step 1** is to take a screenshot of a method. Lots of you did not scroll down enough to see any test cases thus.

TDD using JUnit (JAVA) will autogenerate test cases for the class you specify, in your case the AddressBookType Class that will have 8 methods. At this stage you are to write the code for the test cases but not have code for the AddressBookType Class 8 methods (they will only be skeletons).

NOTE: JUnit has a default fail statement. You must remove it or replace it with your code for that test case, just as presented in class.





You must write the code and the tests fail because there test case fails and not because of the fail statement.

- **Step 2** is not to take a screenshot but to copy and paste AddressBookType.java Class which will contain the methods skeletons and it will have your 4 lines at the beginning of the .java file.

- **Step 3** is not to take a screenshot but to copy and paste AddressBookTypeTest.java Class which will contain the test methods coded and it will have your 4 lines at the beginning of the .java file.

TEAM 50A&S

UML Use Case 100 point

"Requirements" 147 lines? +10

"Step 1, 2" UC

Compilable UML +30

"Step 3" USE CASE DIAGRAM

"Cut" & "Paste"? Word? +30

"Step 4" USE CASE DESCRIPTION

Template +30

UML compile error → stop

All or nothing! MAJOR Fault!

147 lines ✓ line 1 ? -10

100

Berkowitz
Liang
Jaradi

Not assigned work?!

TEAM50AGS

SE Team Project with Line Numbers
TEXTUAL ANALYSIS for
Requirements Workflow
UML USE CASE DIAGRAM

# of Actors: 4
Table of Actors:

A1	*Owner*
A2	*Salesperson*
A3	*Customer*
A4	*DBA*
A5	*Guest*
A6	*Customer Service Operator*
A7	*Worker*

# of UCs: 14
Table of UCs:

UC1	*request customer report*
UC2	*request Artist report*
UC3	*request customer artist preferences report*
UC4	*purchase new art*
UC5	*repurchase art from customer*
UC6	*request past purchase report*
UC7	*request past purchase artwork location report*
UC8	*request artist and works report*
UC9	*request speed of sale report*
UC10	*request current inventory report*
UC11	*modify account*
UC12	*create account*
UC13	*add new artist*
UC14	*add new customer*

UC15	request online chat
UC16	request search
UC17	request art framing service
UC18	request work order report
UC19	log in
UC20	Sell art

Version 2.0

1

line 1?

some reason. Search engines spiders also latch on to this content, especially if the image is linked to another page. 'alt' text will also pop up when a visitor moves their mouse over the image. Client requires that pictures and (not required) videos be used in **TEAM5OAGS.**

- **TEAM5OAGS** site navigation should be simple and all the questions a consumer may ask should be answered along the way. Where possible, adhere to the "three click rule" - that is, a visitor should be able to access any information regarding your service within 3 clicks of any other area of your web site.

- **TEAM5OAGS** should take advantage of the Javascript and Ajax technologies.

UC15: **request online chat**

"C" complete?

- **TEAM5OAGS** implements a live online support chat allowing communication of

A5: Guest A3: Customer A6: Customer Service Operator

guests and customers with **TEAM5OAGS** operators over the Internet in real time directly from **TEAM5OAGS** web site. No chat information will be stored in the database.

- **TEAM5OAGS** web site will have at least the following pages "**TEAM5OAGS**" home page, "About Us" page, "Contact Us" page, "Testimonials" page with a

Team Project: TEAM5OAGS (OnLine Art Gallery System) Web Site

The **TEAM5OAGS** web site should be designed using these principles:

- Text must be grammatically sound and spelled correctly. Poor spelling loses credibility points straight away. Ensure that there is plenty of well laid out textual content on the site to attract search engines as well as to inform prospective clients.
- Use keyword and key phrase rich text; that is, utilize copy that includes common phrases that people would enter into search engines when performing a query.
- **TEAM5OAGS** needs to be viewable from at least IE and Firefox browsers.
- Images are a wonderful medium to assist in the online application, especially useful to those clients with poor literacy levels or who are in a rush, as we all seem to be these days. But remember, while a picture may be worth a thousand words in the offline world, its worth next to nothing when it comes to search engines as spiders do not 'see' pictures.

  Image HTML coding should also contain 'alt' tags. This is a textual representation of the image which is useful for the situations where the image doesn't load for

UC16: **request search**

possible search function. **Testimonials are stored in the database. The search is done by a stored procedure.**

- **TEAM5OAGS** is a secure web application with a relational database backend (**TEAM5OAGS**).

UC20: **sell art**

TEAM5 is a small art gallery that sells contemporary European and North American fine art, including lithographs, high-quality reproduction prints, original paintings and other artwork, and photographs. All of the lithographs, prints, and photos are signed and numbered, and the original art is usually signed. **TEAM5** also provides art framing

UC17: **request art framing service**

services. It creates a custom frame for each artwork (rather than selling standardized, premade frames) and is known for its excellent collection of frame stock.

TEAM5 emphasizes reproduction artworks of European Impressionist, Abstractionist, and Modernist artists such as Wassily Kandinsky and Henri Matisse. For original art, **TEAM5** concentrates on Northwest School artists, such as Mark Tobey, Morris Graves,

GuyAnderson, and Paul Horiuchi, and produces shows of contemporary artists who work in the Northwest School tradition or in Northwest Maritime art. The price of new reproduction prints ranges up to $1,000, and prices for contemporary artists range from $500 to $10,000. The price of art from the Northwest School artists varies considerably, depending on the artwork itself. Small pencil, charcoal, or watercolor sketches may sell for as little as $2,000, whereas major works can range from $10,000 to $100,000. Very occasionally, **TEAM5** may carry Northwest School art priced up to $500,000, but art priced above $250,000 is more likely to be sold at auction by a major art auction house.

TEAM5 has been in business for 30 years and has one full-time owner, three salespeople, and two workers who make frames, hang art in the gallery, and prepare artwork for shipment.

A7: Worker

UC18: **request work order report**

TEAM5 holds openings and other gallery events to attract customers to the gallery.

TEAM5 owns all of the art that it sells—even sales of contemporary artwork is treated as a purchase by **TEAM5** that then is resold to a customer. **TEAM5** does not take items on a consignment basis.

The requirements for the **TEAM5** application are as follows:

A1: Owner A2: Salesperson

First, both the owner and the salespeople want to keep track of Customers' last and first names, addresses (street, city, state, zip, country), phone numbers (area code and phone number), and e-mail addresses by requesting a Customers Report and want to keep track of Artists' last and first names and nationality by requesting an Artists Report. They also want to know which artists have appeal to which customers by requesting a Customer Artists Preferences Report where the Customer Id is the input to such report. The salespeople use this information to determine whom to contact when new art arrives and to personalize verbal and e-mail communications with their customers.

UC1: request customer report

UC2: request Artist report

UC3: request customer artist preferences report

UC4: purchase new art

When TEAM5 purchases new art, data about the artist, the nature of the work, the acquisition date, and the acquisition price are recorded in the database backend. Also, on occasion, TEAM5 repurchases art from a customer and resells it, thus a work may appear in the TEAM5 gallery multiple times.

UC5: repurchase art from customer

When art is repurchased, the artist and work data are not reentered, but the most recent acquisition date and price are recorded.

There is a policy at TEAM5 to set the value of AskingPrice equal either to twice the AcquisitionPrice or to the AcquisitionPrice plus the average net gain for sales of this art

in the past, whichever is greater. An AFTER database trigger **TRANSACTION_AskingPriceInitialValue** is to implement this policy. After declaring program variables, the trigger reads the TRANSACTION table to find out how many TRANSACTION rows exist for this work. Because this is an AFTER trigger, the new TRANSACTION row for the work will have already been inserted. Thus, the count will be one if this is the first time the work has been in the gallery. If so, the new value of SalesPrice is set to twice the AcquisitionPrice.

If the user variable rowCount is greater than one, then the work has been in the gallery before. To compute the average gain for this work, the trigger uses an **ArtistWorkNetView** view to compute SUM(NetProfit) for this work. The sum is placed in the variable sumNetProfit. Notice that the WHERE clause limits the rows to be used in the view to this particular work. The average is then computed by dividing this sum by rowCount minus one.

In addition, when art is sold a TRANSACTION record with the purchase date, acquisition price, date acquired, sales price, asking price, and identity of the purchasing customer are stored in the database backend.

TEAM5 has a special interest in Mexican painters and never discounts the price of their works. Thus, the SalesPrice of a work must always be at least the AskingPrice. To enforce this rule, TEAM5 database has an insert and update trigger TRANSACTION_CheckSalesPrice on TRANSACTION that checks to see if the work is by a Mexican painter. If so, the SalesPrice is checked against the AskingPrice. If it is less than the AskingPrice, the SalesPrice is reset to the AskingPrice. This, of course, must happen when the art work is actually being sold, and the customer charged the full amount!

Salespeople want to examine past purchase data so that they can devote more time to the most active buyers by requesting a Past Purchase Report. They also sometimes use the purchase records to identify the location of artworks they have sold in the past by requesting a Past Purchases Artwork Location Report.

UC6: request past purchase report

UC7: request past purchase artwork location report

TEAM5 wants to hire you to create TEAM5OAGS web application for these requirements.

For marketing purposes, TEAM5 wants its TEAM5OAGS web application to provide a

UC8: request artist and works report

list of artists and works that have appeared in the gallery by requesting an Artist and Works Report. The owner also would like to be able to determine how fast an artist's

UC9: request speed of sale report

work sells and at what sales margin by requesting a Speed of Sale Report given an artist Id. The TEAM5OAGS web application also should display current inventory on a web

UC10: request current inventory report

page that customers can access via the Internet by requesting a Current Inventory Report.

UC19: log in

TEAM5OAGS is a secure web application where owner, sales people, customers have

UC11: modify account

accounts and have the ability to modify their own information (not the USERNAME,

UC12: create account

PASSWORD, EMAIL ADDRESS which will be assigned by the TEAM5OAGS web

A4: DBA

application manager/DBA. DBA has access to New Artist Forms and New Customer

UC13: add new artist

Forms (it should include the artist's interested nationality) to allow them to add new

UC14: add new customer

artists and customers.

The contract states that, at the acceptance test, a scenario of usage of the **TEAM5OAGS** will be given to you. **Data about customers, art work, artists, etc. will be given to the DBAs for your TEAM5.**

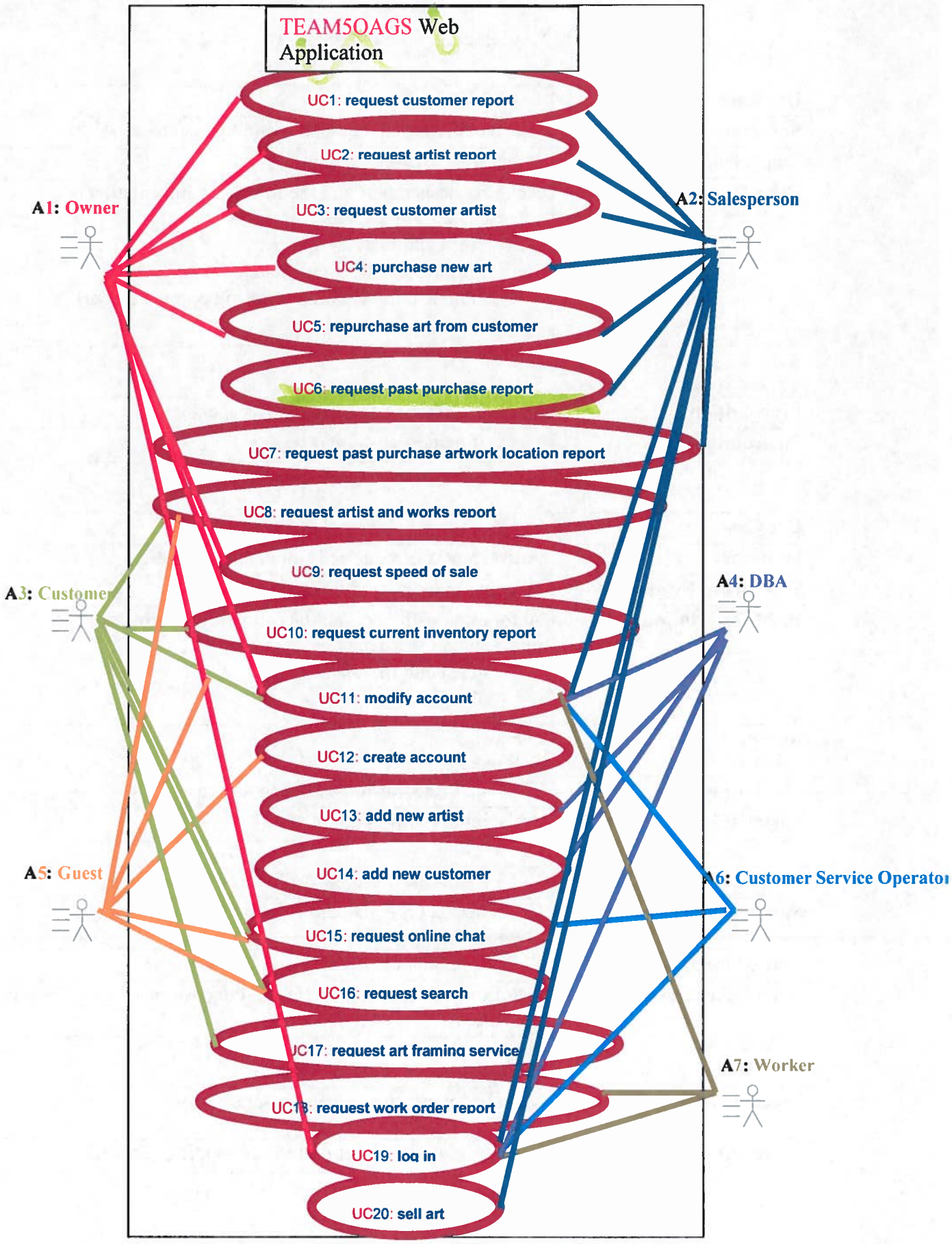
The client desires that **TEAM5OAGS** be designed and implemented using OO paradigm. Formal Analysis, SRS (Software Requirements Specification) document needs to be signed by the **TEAM5** and the client BEFORE any design is started. A SPMP (Software Project Management Plan) document needs to be delivered before the actual development is started.

A prototype is highly desirable. Graphical User Interfaces, Web Site Design, each Page Design can expedite the development process.

Coordinate with the database backend DBAs in clearly separating what is going to be placed in the database and what will be implemented in the database (backend) versus what will be implemented in the **TEAM5OAGS** web application (front end).

An MVC (3 tier architecture) is required.

NOTE: This requirements document "SE Team Project with Line Numbers.doc" might go through revisions based on your questions. There might be missing requirements; there might be unclear requirements, or conflicting requirements. I can take 5 minutes each class to answer possible questions.


TEAM5OAGS Web Application
UC1: request customer report
UC2: request artist report
UC3: request customer artist
UC4: purchase new art
UC5: repurchase art from customer
UC6: request past purchase report
UC7: request past purchase artwork location report
UC8: request artist and works report
UC9: request speed of sale
UC10: request current inventory report
UC11: modify account
UC12: create account
UC13: add new artist
UC14: add new customer
UC15: request online chat
UC16: request search
UC17: request art framing service
UC18: request work order report
UC19: log in
UC20: sell art
A1: Owner
A2: Salesperson
A3: Customer
A4: DBA
A5: Guest
A6: Customer Service Operator
A7: Worker

UML Use Case Descriptions

Use Case	UC 1 – Request Customers Report
Summary	Customer report is requested from the database
Triggering Event	Request for customer records
Brief Description	Actor requests report and the following information is represented from database • Last and First names • Addresses(street,city,state,zip,country) • Phone numbers(area code and phone number) • Email addresses
Actors	Owner Salesperson
Precondtions	Must be logged in as owner or salesperson
Postconditions	View of request customers report

Use Case	UC 2 – Request Artists Report
Summary	Artists report is requested from the database
Triggering Event	Wanting to look at Artist records
Brief Description	Actor requests report and the following information is represented from database • Last and first names • Nationality
Actors	Owner Salesperson
Precondtions	The Artist Records must exist in database
Postconditions	Artist report view is returned

Use Case	UC 3 – Request customer artist preferences report
Summary	Customer report is requested from the database based on previous bought artwork
Triggering Event	Request to look at customer artist preferences
Brief Description	Actor inputs the customer ID and the customers art preferences are listed
Actors	Owner Salesperson
Precondtions	Must be logged in as owner or salesperson
Postconditions	View or customer artist preference report is retuned

Use Case	UC 4 – Purchase New Art
Summary	A new piece of art is purchased and added to the inventory
Triggering Event	New art is acquired
Brief Description	The following information is input when buying new art. • Data about the artist • Nature of the work • Acquisition date • Acquisition price All input into database
Actors	Owner Salesperson
Precondtions	Must be logged in as owner or salesperson
Postconditions	Purchase new art view is returned

Use Case	UC 5 – Repurchase Art From Customer
Summary	A piece of art is repurchased and added back to the inventory
Triggering Event	New art is acquired
Brief Description	The following information is input when buying back the art. • Data about the artist • Nature of the work • Acquisition date • Acquisition price All input into database
Actors	Owner Salesperson
Precondtions	Must be logged in as owner or salesperson
Postconditions	Repurchased view is returned

Use Case	UC 6 – Request Past Purchase Report
Summary	Report to examine past purchase data so that actor can devote more time to the most active buyers
Triggering Event	Trying to sell more art
Brief Description	A report displaying the past purchases
Actors	Salesperson
Precondtions	Must be logged in as salesperson
Postconditions	Past purchase view is returned

Use Case	UC 7 – Request Past Purchase Artwork Location Report

Summary	To identify the location of artworks they have sold in the past
Triggering Event	Trying to sell more art
Brief Description	A report displaying the location of past purchased artwork
Actors	Salesperson
Precondtions	Must be logged in as salesperson
Postconditions	Past purchase Artwork Location report view is returned

Use Case	UC 8 – Request Artist and Works Report
Summary	To provide a list of artists and works that have appeared in the gallery
Triggering Event	Trying to sell more art
Brief Description	A report displaying each artist and the works he/she has done.
Actors	Customer Guest
Precondtions	Must be logged in as customer or guest
Postconditions	Artist and works view is returned

Use Case	UC 9 – Request Speed of Sale Report
Summary	To determine how fast an artist's work sells and at what sales margin
Triggering Event	Request is made to site
Brief Description	Given and artists id a report is given based on what has sold and how much profit
Actors	Owner
Precondtions	Must be logged in as owner
Postconditions	Speed of sale report view is returned

Use Case	UC 10 – Request Current Inventory Report
Summary	Display current inventory on a web page
Triggering Event	Trying to sell more art
Brief Description	Display current inventory on webpage
Actors	Customer
Precondtions	Must be logged in as customer
Postconditions	Current instance of all inventory is added to inventory view and returned to user

Use Case	UC 11 – Modify Account
Summary	All actors have accounts and have the ability to modify their

	email or password
Triggering Event	Need to change info
Brief Description	Modify view is retuned to change password, email or both.
Actors	Owner SalesPerson Customer
Precondtions	Account info must exist for Actor and Must be logged in
Postconditions	Account info in modified

Use Case	UC 12 – Create Account
Summary	Forms to fill out to assign account information
Triggering Event	New Account information is sent to DBA's view to add new customer or not
Brief Description	All relevant information about account holder in assigned
Actors	Guest
Precondtions	Must not be logged in
Postconditions	Registration view is returned

Use Case	UC 13 – Add New Customer
Summary	New customer information is added to the database
Triggering Event	A new customer is acquired
Brief Description	Customer information is entered into the database • Last and First names • Addresses(street,city,state,zip,country) • Phone numbers(area code and phone number) Email addresses
Actors	DBA
Precondtions	Guest must have already used create an account & Must be logged in as DBA
Postconditions	New customer is added to the database

Use Case	UC 14 – Add New Artist
Summary	New Artist information is added to the database
Triggering Event	A new Artist is acquired
Brief Description	Customer information is entered into the database • Last and first names • Nationality

Actors	DBA
Precondtions	Customer database must exist & Must be logged in as DBA
Postconditions	New customer is added to the database

Use Case	UC 15 – Request Online Chat
Summary	Customer or Guest chats with customer service operator
Triggering Event	Guest or customer request to chat with operators
Brief Description	• Customer service operator chats with guest or customer to answer any questions they might have
Actors	Customer or Guest
Precondtions	Customer database must exist and must be logged in as DBA
Postconditions	Customer or Guest connects with first available customer servie operator

Use Case	UC 16 – Request Search
Summary	Search keys are entered into the search text box and user requests search (Integrated Custom Google Search)
Triggering Event	Customer or guest requests to search for keys in website
Brief Description	• Customer or guest can request a search for particular keys on the website which returns all relevant pages.
Actors	Customer or Guest
Precondtions	Must be logged in as customer or guest
Postconditions	Search results page is returned

Use Case	UC 17 – Request Art Frame Service
Summary	Customer orders frame from the frame stock
Triggering Event	
Brief Description	Customer information is entered into the database • Last and first names • Nationality
Actors	Customer
Precondtions	Customer must be on frame order page and have appropriate input data filled
Postconditions	An order for frame stock is placed with measurements and added to the work order list

Use Case	UC 18 – Request Work Order List
Summary	Worker sees orders that need to be completed
Triggering Event	Worker logged in

Brief Description	• Worker completes current work orders in work order list such as building frames for art orders, hanging art, and shipping new orders.
Actors	Worker
Precondtions	Worker must be logged in
Postconditions	After worker checks that an order is complete it is then removed from the work order list

Use Case	UC 19 – Log in
Summary	Guest logged into website and verifies account
Triggering Event	Username and password is verified
Brief Description	Guest logged into website, username and password is first verified then privileges and home view is returned based on account type
Actors	Guest
Precondtions	Guest must request to log in
Postconditions	If verified account's type view is returned else incorrect password view is returned with chance to register or recovery user name or password

Use Case	UC 20– sell art
Summary	Salesperson uses sell art to process an order for a customer
Triggering Event	Salesperson process order
Brief Description	Salesperson fills out an order form for a customer then hits the process order button.
Actors	Salesperson
Precondtions	Salesperson must be logged in and order form filled out
Postconditions	Order is placed in the system and additional information is sent to work order list

DOCUMENT CONTROL

CHANGE HISTORY

TLs entries (assigned work and due dates) before releasing to the team

Revision	Name	Due Date	Description
1.1	Lee, Ryan	02/01/2013	Identified Actors and Use Cases
1.2	Loucks, Joel	02/01/2013	Complete Use Case Modeling
1.3	Ohlson, Gabriel	02/01/2013	Complete Use Case Modeling
1.X	Cruz, Christopher	02/05/2013	Review Document
1.Y	Naviwala, Muhammad	02/05/2013	Review Document

TMs entries when they completed their work

Revision	Name	**Completed Date**	Description
1.A	Joel Loucks	01/20/2013	Identified actors for use case diagram.
1.B	Ryan Lee	01/20/2013	Identified additional actors and Use Cases.
1.C	Ryan Lee	01/20/2013	Added preliminary UML UC diagram.
1.D	Joel Loucks	01/20/2013	Made actors in diagram a group and added UC# labels to UC diagram.
1.E	Muhammad Naviwala	01/22/2013	Reviewed document. Changed/completed UC diagram. Changed some grammar in sentences.
1.F	Christopher Cruz	01/22/2013	Reviewed, edited some grammar. Added class diagrams with just attributes. Still missing class diagram.
1.G	Joel Loucks	01/24/2013	Added actors and use cases to their respective tables on page 1 of document.
1.H	Ryan Lee	01/24/2013	Finished preliminary text analysis and UC diagram.
1.I	Christopher Cruz	01/24/2013	Cleaned up tables on first page.
1.J	Joel Loucks	01/25/2013	Added use case descriptions for each use case, but still need some info added in some of them. Added line from customer to UC10.

1.K	Joel Loucks	01/28/2013	Updated my work in the change history in the word document.
1.L	Gabriel Ohlson	01/28/2013	Added lots of color and changed some use cases and use case descriptions.
1.M	Christopher Cruz	01/29/2013	Final review and sign.
1.N	Muhammad Naviwala	01/30/2013	Document looks pretty good and the new revision has followed all guidelines and templates. Final review done.
1.O	Gabriel Ohlson	02/01/2013	Found 3 more actors and added 5 more use cases. Updated all use cases and filled out all left over boxes as well as changed them to account for new use cases
1.P	Cruz Christopher	02/01/2013	Fixed some formatting, reviewed. Really looking good now.
1.Q	Muhammad Naviwala	02/05/2013	Modified the aligning of the tables and some formatting. The use case numbering is not in ordering though. Everything looks okay now. Final review done.
1.R	Cruz, Christopher	02/05/2013	Final Review done, looks great.
1.S	Ryan Lee	02/05/2013	Team leader final review and approval for Red Deliverables.

TLs entry before RED TEAMS DELIVERABLES

Revision	Name	Due Date	Description
2.0	Lee, Ryan	02/05/2013	I changed Version to 2.0

DOCUMENT STORAGE

This file is stored in SVN at http://limi.cs.uh.edu/COSC4351/TEAM5/TEAM PROJECT DELIVERABLES/**SE Team Project with Line Numbers for UML USE CASE DIAGRAM.doc**.

team 5

TEAM SOAGS

UML Class Diagram

100 points

"Requirements"

147 lines ?

+10

"Step 1, 2"

Compilable UML

+50

"Step 3"

OOA only! -30

"Step 4"
CLASS DIAGRAM
MVC

"Cut" & "Paste" ?

Word ?

+40

Used USE CASE ? -?10

Actor Controller ?

All or nothing! MAJOR fault!

UML compile error, slap

147 lines? (60)

Loucks
Ohlson

not assigned work!!

TEAM5OAGS

SE Team Project with Line Numbers
TEXTUAL ANALYSIS for
OOA Workflow
UML CLASS DIAGRAM

# of Classes: ***14***

Table of Classes:

Class #	***Class Name***
1	AccountHolder
2	Artist
3	Artwork
4	Customer
5	Transaction
7	Testimonial
8	Frame
9	Inventory
10	ReportController
11	OrderController
12	AccountController
13	DBAController
14	ChatController
15	Model
16	View
17	Controller

# of Attributes: 42

Table of Attributes:

Class Name	***Attributes***
Artwork	artID natureWork artImage
AccountHolder	last first street city state zip country areaCode phoneNumber email role userID password
Artist	aLast aFirst nationality artistID
Customer	CID
Transaction	acquisitionDate acquisitionPrice purchaseDate salesPrice askingPrice location transID
Testimonial	testID Test_Content Test_Date
Employee	EID
Frame	FID frameType width height

	rabbet color style
Inventory	itemID itemType itemPrice

# of Methods : 20

Table of Methods:

Class Name	Methods
ReportController	requestCustomersReport requestArtistsReport requestCustomerArtistPreferencesReport requestPastPurchaseReport requestPastPurchasesArtworkLocationReport requestArtistandWorksReport requestSpeedofSaleReport requestCurrentInventoryReport requestSearch requestWorkOrderReport
OrderController	purchaseNewArt repurchaseArt requestArtFramingService sellArt
AccountController	modifyAccount loginAccount
DBAController	createAccount addArtist addCustomer
ChatController	requestOnlineChat

Version 2.0

1

Steps 1, 2, 3 - Identify Classes, Attributes, verbs Methods
(Square = Class, Oval = Attributes, Underline = verb Methods)

Team Project: TEAM5OAGS (OnLine Art Gallery System) Web Site

The TEAM5OAGS web site should be designed using these principles:

- Text must be grammatically sound and spelled correctly. Poor spelling loses credibility points straight away. Ensure that there is plenty of well laid out textual content on the site to attract search engines as well as to inform prospective clients.
- Use keyword and key phrase rich text; that is, utilize copy that includes common phrases that people would enter into search engines when performing a query.
- TEAM5OAGS needs to be viewable from at least IE and Firefox browsers.
- Images are a wonderful medium to assist in the online application, especially useful to those clients with poor literacy levels or who are in a rush, as we all seem to be these days. But remember, while a picture may be worth a thousand words in the offline world, its worth next to nothing when it comes to search engines as spiders do not 'see' pictures.

  Image HTML coding should also contain 'alt' tags. This is a textual representation of the image which is useful for the situations where the image doesn't load for

some reason. Search engines spiders also latch on to this content, especially if the image is linked to another page. 'alt' text will also pop up when a visitor moves their mouse over the image. Client requires that pictures and (not required) videos be used in TEAM5OAGS.

- TEAM5OAGS site navigation should be simple and all the questions a consumer may ask should be answered along the way. Where possible, adhere to the "three click rule" - that is, a visitor should be able to access any information regarding your service within 3 clicks of any other area of your web site.

- TEAM5OAGS should take advantage of the Javascript and Ajax technologies.

ChatController: 15 - Method: requestOnlineChat

- TEAM5OAGS implements a live online support chat allowing communication of guests and customers with TEAM5 OAGS operators over the Internet in real time directly from TEAM5OAGS web site. No chat information will be stored in the database.

Customer

- TEAM5OAGS web site will have at least the following pages "TEAM5OAGS" home page, "About Us" page, "Contact Us" page, "Testimonials" page with a

Testimonial

ReportController: 16 - Method: requestSearch

possible search function. Testimonials are stored in the database. The search is done by a stored procedure.

- TEAM5OAGS is a secure web application with a relational database backend (TEAM5OAGS).

TEAM5 is a small art gallery that sells contemporary European and North American fine art, including lithographs, high-quality reproduction prints, original paintings and

Artwork

other artwork, and photographs. All of the lithographs, prints, and photos are signed and

Artwork: 1 - artID

numbered and the original art is usually signed. TEAM5 also provides art framing

OrderController: 17 - Method: requestArtFramingService

services. It creates a custom frame for each artwork (rather than selling standardized,

Frame

premade frames) and is known for its excellent collection of frame stock.

TEAM5 emphasizes reproduction artworks of European Impressionist, Abstractionist,

Artist

and Modernist artists such as Wassily Kandinsky and Henri Matisse. For original art, TEAM5 concentrates on Northwest School artists, such as Mark Tobey, Morris Graves,

49 GuyAnderson, and Paul Horiuchi, and produces shows of contemporary artists who work

50 in the Northwest School tradition or in Northwest Maritime art. The price of new

51 reproduction prints ranges up to $1,000, and prices for contemporary artists range from

52 $500 to $10,000. The price of art from the Northwest School artists varies considerably,

53 depending on the artwork itself. Small pencil, charcoal, or watercolor sketches may sell

54 for as little as $2,000, whereas major works can range from $10,000 to $100,000. Very

55 occasionally, TEAM5 may carry Northwest School art priced up to $500,000, but art

56 priced above $250,000 is more likely to be sold at auction by a major art auction house.

Employee

57 TEAM5 has been in business for 30 years and has one full-time owner, three salespeople

ReportController: 18 - Method: requestWorkOrderReport

58 and two workers who make frames, hang art in the gallery, and prepare artwork for

59 shipment.

60 TEAM5 holds openings and other gallery events to attract customers to the gallery.

61 TEAM5 owns all of the art that it sells—even sales of contemporary artwork is treated

62 as a purchase by TEAM5 that then is resold to a customer. TEAM5 does not take items

63 on a consignment basis.

64 The requirements for the TEAM5 application are as follows:

AccountHolder: 2 - first
AccountHolder: 1 - last

65 First, both the owner and the salespeople want to keep track of Customers last and first

AccountHolder: 3 - street, 4 - city, 5 - state, 6 - zip, 7 - AccountHolder: 8 - areaCode

66 names, addresses (street, city, state, zip, country), phone numbers (area code and phone

AccountHolder: 1 - email
AccountHolder: 9 - phoneNumber
ReportController: 1 - Method: requestCustomersReport

67 number), and e-mail addresses by requesting a Customers Report and want to keep track

Artist: 1 - aLast
Artist: 2 - aFirst
Artist: 3 - nationality
ReportController: 2 - Method: requestArtistsReport

68 of Artists (last and first) names and nationality by requesting an Artists Report. They

69 also want to know which artists have appeal to which customers by requesting a

Customer: 10 – CID
Controller: 3 - Method: requestCustomerArtistsPreferencesReport

70 Customer Artists Preferences Report where the Customer Id is the input to such report.

71 The salespeople use this information to determine whom to contact when new art arrives

72 and to personalize verbal and e-mail communications with their customers.

Artwork: 2 - natureWork
OrderController: 4 - Method: purchaseNewArt

73 When TEAM5 purchases new art, data about the artist, the nature of the work, the

Transaction: 1 - acquisitionDate
Transaction: 2 - acquisitionPrice

74 acquisition date, and the acquisition price are recorded in the database backend. Also, on

OrderController: 5 - Method: repurchaseArt

75 occasion, TEAM5 repurchases art from a customer and resells it, thus a work may appear

76 in the TEAM5 gallery multiple times.

77 When art is repurchased, the artist and work data are not reentered, but the most recent

78 acquisition date and price are recorded.

79 There is a policy at TEAM5 to set the value of AskingPrice equal either to twice the

80 AcquisitionPrice or to the AcquisitionPrice plus the average net gain for sales of this art

in the past, whichever is greater. An AFTER database trigger

TRANSACTION_AskingPriceInitialValue is to implement this policy. After declaring

Transaction

program variables, the trigger reads the TRANSACTION table to find out how many

TRANSACTION rows exist for this work. Because this is an AFTER trigger, the new

TRANSACTION rowfor the work will have already been inserted. Thus, the count will

be one if this is the first time the work has been in the gallery. If so, the new value of

SalesPrice is set to twice the AcquisitionPrice.

If the user variable rowCount is greater than one, then the work has been in the gallery

before. To compute the average gain for this work, the trigger uses an

ArtistWorkNetView view to compute SUM(NetProfit) for this work. The sum is placed

in the variable sumNetProfit. Notice that the WHERE clause limits the rows to be used in

the view to this particular work. The average is then computed by dividing this sum by

rowCount minus one.

OrderController: 20 - Method: sellArt Transaction: 3 - purchaseDate

In addition, when art is sold a TRANSACTION record with the purchase date,

Transaction: 4 - salesPrice Transaction: 5 - askingPrice

acquisition price, date acquired sales price asking price and identity of the purchasing

customer are stored in the database backend.

97 TEAM5 has a special interest in Mexican painters and never discounts the price of their

98 works. Thus, the SalesPrice of a work must always be at least the AskingPrice. To

99 enforce this rule, TEAM5 database has an insert and update trigger

100 TRANSACTION_CheckSalesPrice on TRANSACTION that checks to see if the work

101 is by a Mexican painter. If so, the SalesPrice is checked against the AskingPrice. If it is

102 less than the AskingPrice, the SalesPrice is reset to the AskingPrice. This, of course, must

103 happen when the art work is actually being sold, and the customer charged the full

104 amount!

105

106 Salespeople want to examine past purchase data so that they can devote more time to the

ReportController: 6 - Method: requestPastPurchaseReport

107 most active buyers by requesting a Past Purchase Report. They also sometimes use the

Transaction: 6 - location

108 purchase records to identify the location of artworks they have sold in the past by

ReportController: 7 - Method: requestPastPurchasesArtworkLocationReport

109 requesting a Past Purchases Artwork Location Report.

110 TEAM5 wants to hire you to create TEAM5OAGS web application for these

111 requirements.

112 For marketing purposes, TEAM5 wants its TEAM5OAGS web application to provide a

ReportController: 8 - Method: requestArtistandWorksReport

113 list of artists and works that have appeared in the gallery by requesting an Artist and

114 Works Report. The owner also would like to be able to determine how fast an artist's

ReportController: 9 - Method: requestSpeedofSaleReport

115 work sells and at what sales margin by requesting a Speed of Sale Report given an artist

Artist: 4 - artistID

116 Id. The TEAM5OAGS web application also should display current inventory on a web

ReportController: 10 - Method: requestCurrentInventoryReport

117 page that customers can access via the Internet by requesting a Current Inventory

118 Report.

AccountHolder: 2 - role

AccountController: 19 - Method: loginAccount

119 TEAM5OAGS is a secure web application where owner, sales people, customers have

Accountholder

AccountHolder: 3 - userID

AccountController: 11 - Method: modifyAccount

120 accounts and have the ability to modify their own information (not the USERNAME,

AccountHolder: 4 - password

DBAController: 12 - Method: createAccount

121 PASSWORD, EMAIL ADDRESS which will be assigned by the TEAM5OAGS web

122 application manager/DBA. DBA has access to New Artist Forms and New Customer

DBAController: 13 - Method: addNewArtist

123 Forms (it should include the artist's interested nationality) to allow them to add new

DBAController: 14 - Method: addNewCustomer

124 artists and customers.

125

The contract states that, at the acceptance test, a scenario of usage of the **TEAM5OAGS** will be given to you. **Data about customers, art work, artists, etc. will be given to the DBAs for your TEAM5.**

The client desires that **TEAM5OAGS** be designed and implemented using OO paradigm. Formal Analysis, SRS (Software Requirements Specification) document needs to be signed by the **TEAM5** and the client BEFORE any design is started. A SPMP (Software Project Management Plan) document needs to be delivered before the actual development is started.

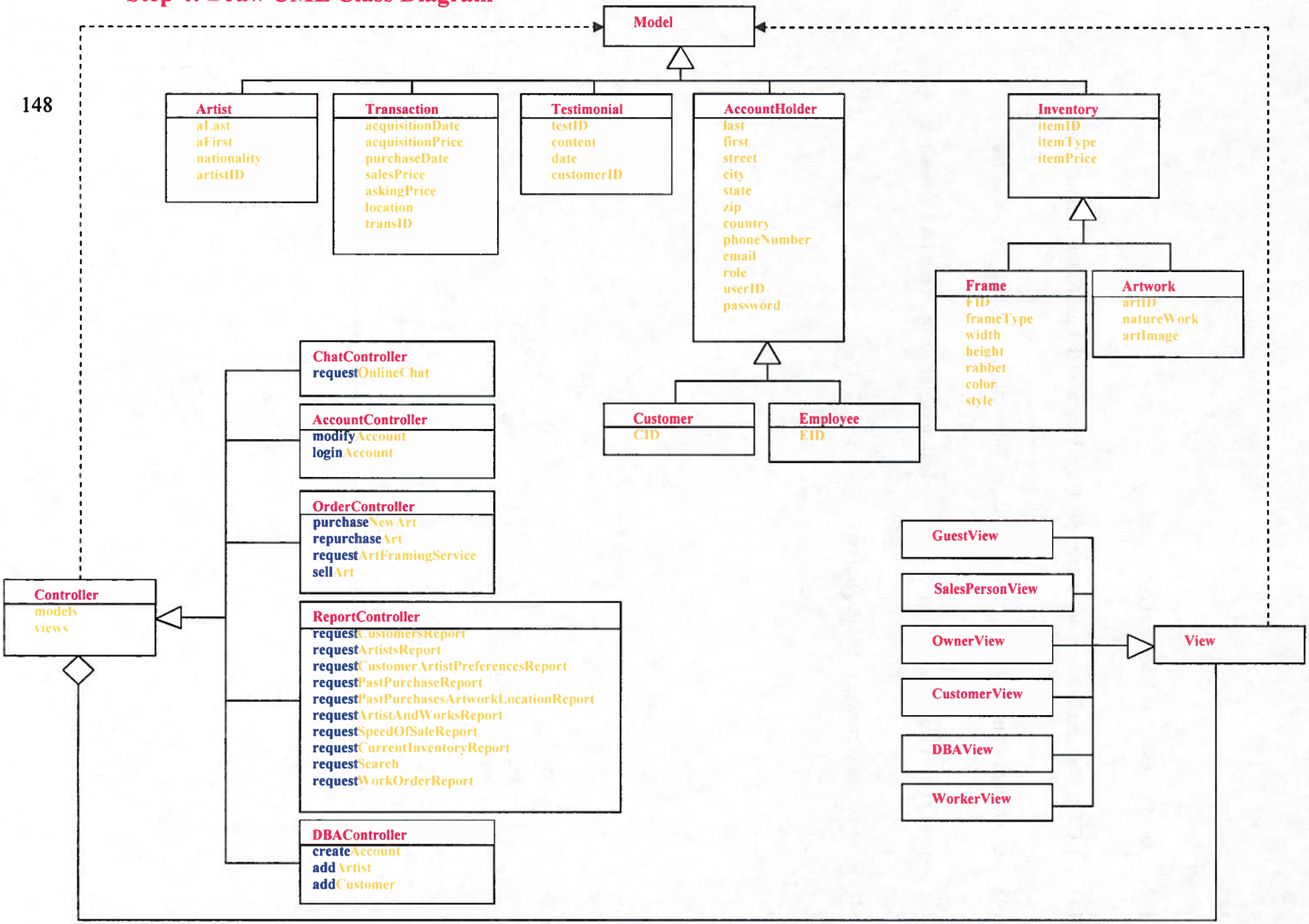
A prototype is highly desirable. Graphical User Interfaces, Web Site Design, each Page Design can expedite the development process.

Coordinate with the database backend DBAs in clearly separating what is going to be placed in the database and what will be implemented in the database (backend) versus what will be implemented in the **TEAM5OAGS** web application (front end).

An MVC (3 tier architecture) is required.

NOTE: This requirements document "SE Team Project with Line Numbers.doc" might go through revisions based on your questions. There might be missing requirements; there might be unclear requirements, or conflicting requirements. I can take 5 minutes each class to answer possible questions.

Step 4: Draw UML Class Diagram



Step 5 – Detailed Design Pseudocode

Method	**ReportController**: 1 - **Method: requestCustomersReport**
Description	Display a report listing Customers' Information
Pseudocode	If Login is Owner or Salesperson Request from Customer to get: • Last and First names • Addresses(street,city,state,zip,country) • Phone numbers(area code and phone number) • Email addresses Return view of request report Else Reject request
Parameters	userID

Method	**ReportController**: 2 - **Method: requestArtistsReport**
Description	Display a report listing Artists' Information.
Pseudocode	If Login is Owner or Salesperson Request from Artist to get: • Last and First names of Artist • Nationality Return view of request report Else Reject request
Parameters	userID

Method	**ReportController**: 3 - **Method: requestCustomerArtistsPreferencesReport**
Description	Display a report in which artists have appeal to which customers.
Pseudocode	If Login is Owner or Salesperson Request from Artist to get: • Last and first names of Artist WHERE Input CID = CID of Transaction AND artID of Transaction = artID of Artwork AND artistID of Artwork = artistID of Artist Return view of request customers report Else Reject request
Parameters	userID, CID

Method	**OrderController**: 4 - **Method: purchaseNewArt**
Description	A new piece of art is purchased and added to the inventory

Pseudocode	If Login is Owner or Salesperson Request from Artist, Artwork & Transaction to set: • Data about the artist (aLast, aFirst, nationality) • Nature of the work • Acquisition date • Acquisition price Else Reject
Parameters	userID, aLast, aFirst, nationality, natureWork, acquisitionDate, acquisitionPrice

Method	**OrderController**: 5 - **Method**: **repurchaseArt**
Description	A piece of art is repurchased and added back to the inventory
Pseudocode	If Login is Owner or Salesperson Request from Transaction to set: • Acquisition date • Acquisition price WHERE Input natureWork = natureWork of Artwork AND artID of Artwork = artID of Transaction Else Reject
Parameters	userID, natureWork, acquisitionDate, acquisitionPrice

Method	**ReportController**: 6 - **Method**: **requestPastPurchaseReport**
Description	Report to examine past purchase data based on recent date of purchase so that actor can devote more time to the most active buyers
Pseudocode	If Login is Salesperson Request from Transaction & Customer to get: • Purchase Date • Last & First Name of Customer • Email Address WHERE CID of Transaction = CID of Customer ORDER BY Purchase Date DESC Return view of request report Else Reject
Parameters	userID

Method	**ReportController**: 7 - **Method**: **requestPastPurchasesArtworkLocationReport**
Description	Report to identify the location of purchased artworks they have sold in the past

Pseudocode	If Login is Salesperson Request from Transaction & Artwork to get: • Transaction ID • Artwork ID • Nature of Work • Location WHERE artID of Transaction = artID of Artwork Return view of request report Else Reject
Parameters	userID

Method	**ReportController**: 8 - **Method: requestArtistandWorksReport**
Description	Report to provide a list of artists and their works that have appeared in the gallery
Pseudocode	If Login is Customer Request from Artist & Artwork to get: • Last and First Name of Artist • Nationality • Artwork ID • Nature of Work WHERE artistID of Artwork = artistID of Artist Return view of request report Else Reject
Parameters	userID

Method	**ReportController**: 9 - **Method: requestSpeedofSaleReport**
Description	Report to determine how fast an artist's work sells and at what sales margin given an artist Id
Pseudocode	If Login is Owner Request from Transaction to create a view with: • Artwork ID • Sales Time = Date of Purchase – Date of Acquisition • Sales Margin = Sales Price – Acquisition Price WHERE artID of Input = artID of Transaction AND purchaseDate != null Return view of request report Else Reject
Parameters	userID, artistID

Method	**ReportController**: 10 - **Method: requestCurrentInventoryReport**
Description	Report to display on web application with current inventory of gallery.

Pseudocode	If Login is Customer Request from Transaction & Artwork to create a view with: • Artwork ID • Nature of Work • artImage • Count based on rowCount in Transaction WHERE purchaseDate of Transaction = null AND artID of Transaction = artID of Artwork Return view of request report Else Reject
Parameters	userID

Method	**AccountController**: 11 - **Method**: **modifyAccount**
Description	Edit Information of their own accounts
Pseudocode	If Login is Customer, SalesPerson, or Owner Request from Customer to set: • Last and First names • Addresses(street,city,state,zip,country) • Phone numbers(area code and phone number) • Email Address • Password
Parameters	userID, last, first, street, city, state, zip, country, areaCode, phoneNumber, email, password

Method	**DBAController**: 12 - **Method**: **createAccount**
Description	Forms to fill out to assign account information
Pseudocode	If Login is DBA Request from AccountHolder to set: • Username and Password for Customer • Email Address • Role of Account (Customer or Employee)
Parameters	userID, username, password, email, role

Method	**DBAController**: 13 - **Method**: **addNewArtist**
Description	New artist information is added to the database
Pseudocode	If Login is DBA Request from Artist to set: • Last and First Name of Artist • Nationality
Parameters	userID

Method	**DBAController**: 14 - **Method**: **addNewCustomer**

Description	New Customer information is added to the database
Pseudocode	If Login is DBA Request from Customer to set: • Last and First names • Addresses(street,city,state,zip,country) • Phone numbers(area code and phone number)
Parameters	userID, last, first, street, city, state, zip, country, areaCode, phoneNumber

Method	**ChatController**: 15 - **Method: requestOnlineChat**
Description	Customer or Guest chats with customer service operator
Pseudocode	Establish a connection between users Accept Input from Customer or Guest or Customer Service Operator Display Input into Chat Box
Parameters	N/A

Method	**ReportController**: 16 - **Method: requestSearch**
Description	Search keys are entered into the search text box and user requests search (Integrated Custom Google Search)
Pseudocode	Accept Input from User Returns all links of relevant pages.
Parameters	N/A

Method	**OrderController**: 17 - **Method: requestArtFramingService**
Description	Service will allow customization for Customer's frame and add it to the frame stock
Pseudocode	If Login is Customer Request from Frame to set: • Type of Material • Width, Height, and Rabbet of Frame • Color • Style Else Reject
Parameters	userID, frameType, width, height, rabbet, color, style

Method	**ReportController**: 18 - **Method: requestWorkOrderReport**
Description	Report to let Worker sees orders that need to be completed
Pseudocode	If Login is Worker Request from Transaction, Frame to get: • Transaction ID • Artwork ID • Frame ID

	• Type of Material • Width, Height, and Rabbet of Frame • Color • Style or Design of Frame Else Reject
Parameters	userID

Method	**AccountController**: 19 - **Method: loginAccount**
Description	Guest logs into website and verifies account if username and/or password is correct
Pseudocode	Initiate Boolean for username Initiate Boolean for password Encrypt input password with key Boolean for username = userID of Input = = UserID of AccountHolder Boolean for password = encrypt password of Input = = encrypt password of AccountHolder WHERE userID of Input = UserID of AccountHolder If Boolean of username AND Boolean for password Return True Else Return False
Parameters	userID, password

Method	**OrderController**: 20 - **Method: sellArt**
Description	Salesperson can sell art to customer after the customer has contacted the Salesperson about purchasing the artwork. Record with the purchase date, acquisition price, date acquired, sales price, asking price, and identity of the purchasing customer are stored.
Pseudocode	If Login is Salesperson Request from Transaction to set: • Purchase and Acquisition Date • Acquisition, Asking, and Sales Price • CID Request from Inventory to set: • Item Type = Art • Item Price = Sales Price Else Reject
Parameters	userID, purchaseDate, acquisitionPrice, acquisitionDate, salesPrice, askingPrice, CID

150 DOCUMENT CONTROL

CHANGE HISTORY

TLs entries (assigned work and due dates) before releasing to the team

Revision	Name	Due Date	Description
1.1	Ryan Lee	02/01/2013	Complete Class Modeling
1.2	Matthew Liang	02/01/2013	Complete Class Modeling
1.3	Yash Jaradi	02/01/2013	Complete Class Modeling
1.X	Christopher Cruz	02/05/2013	Review Document
1.Y	Muhammad Naviwala	02/05/2013	Review Document

TMs entries when they completed their work

Revision	Name	**Completed Date**	Description
1.A	Ryan Lee	01/18/2013	Text analysis update: identified potential objects and attributes for Art, Artist, Customer classes
1.B	Ryan Lee	01/20/2013	Text analysis update: added preliminary UML models
1.C	Christopher Cruz	01/22/2013	Reviewed, edited some grammar. Added class diagrams with just attributes. Still missing class diagram.
1.D	Matthew Liang	01/25/2013	Done TA for steps 1, 2, 3
1.E	Matthew Liang	01/27/2013	Few changes to TA, Added Table Content & UML diagram (step 4)
1.F	Muhammad Naviwala	01/30/2013	Besides the "detailed pseudocode" part, everything looks good. Might need to change template when identifying classes.
1.G	Yash Jaradi	02/01/2013	Cleaned up the TA, changed back to the required templates, found attributes for account holder, added methods for Controller and Transactions.
1.H	Matthew Liang	02/02/2013	Changed TA methods to match Use Case, renumber, bolded text, added class & attributes for frame, reassign classes to attributes
1.I	Yash Jaradi	2/03/2013	Identified possible class Discount and it's attributes and methods and also found 2

			attribute for Transaction which were added to the diagram.
1.J	Matthew Liang	02/03/2013	Added pseudocode for 19 UC methods, Changed attribute Frame_Image to Style, Remove Discount and its property, Reassign rowCount and sumNetProfit to Artwork
1.K	Matthew Liang	02/05/2013	Split the Controller classes into 5 and reassign methods, added Inventory class and its property, add sellArt(), rename a few attribute (look at table), rowCount and sumNetProfit removed.
1.L	Ryan Lee	02/05/2013	Cleaned up the attribute and method tables.
1.M	Yash Jaradi	02/05/2013	Modified the classes according to the new methods for the controller.
1.N	Christopher Cruz	02/05/2013	Connected controllers and views, and reviewed new changes
1.O	Christopher Cruz	02/05/2013	Connected controllers, views, models
1.P	Ryan Lee	02/05/2013	Refined the class diagram in accordance to MVC.
1.Q	Christopher Cruz	02/05/2013	Final Review Done, looks great.
1.R	Muhammad Naviwala	02/05/2013	Modified the formatting of the document a little bit. The class diagram now matches the Table of Attributes. Looks awesome ! Final review done.
1.S	Ryan Lee	02/05/2013	Team leader final review and approval for Red Deliverables

TLs entry before RED TEAMS DELIVERABLES

Revision	Name	Due Date	Description
2.0	Ryan Lee	02/05/2013	I changed Version to 2.0

DOCUMENT STORAGE

This file is stored in SVN at http://limi.cs.uh.edu/COSC4351/~~TEAM5~~/TEAM PROJECT DELIVERABLES/**SE Team Project with Line Numbers for UML CLASS DIAGRAM.doc**.

Team5

100

TEAM5OAGS

Team Project Report

Loucks
Ohlson
Jaradi

No assigned work?

Version 2.0

SVM

Signature Block

COSC 4351	Name	Signature	Date
SE Team Leader	Ryan Lee		2/6/13
SE SQA	Christopher Cruz		2/6/13
SE SQA	Muhammad Naviwala		02/6/2013
SE Team Leader			
SE SQA			
SE SQA			
SE Team Leader			
SE SQA			
SE SQA			

DOCUMENT CONTROL

CHANGE HISTORY

TLs entries (assigned work and due dates) before releasing to the team

Revision	Name	Due Date	Description
1.1	Ryan Lee	02/01/2013	Complete Chapter I
1.2	Matthew Liang	02/01/2013	Complete Chapter II
1.3	Christopher Berkowitz	02/01/2013	Complete Chapter III
1.X	Christopher Cruz	02/05/2013	Review Document
1.Y	Muhammad Naviwala	02/05/2013	Review Document

TMs entries when they completed their work

Revision	Name	**Completed Date**	Description
1.A	Ryan Lee	01/30/2013	Completed all of Chapter I paragraphs.
1.B	Matthew Liang	01/30/2013	Completed Chapter II
1.C	Chris Berkowitz	01/31/2013	Completed chapter 3 part A. More coming shortly.
1.D	Chris Berkowitz	01/31/2013	Completed chapter 3 part B. More coming shortly.
1.E	Chris Berkowitz	01/31/2013	Completed chapter 3 part C. More coming shortly.
1.F	Muhammad Naviwala	02/01/2013	Reviewed Chapter 1, Chapter 2 and part of Chapter 3. Chapter 1 looks fine. For Chapter 2 part A and B, at some places it is hard to understand what you (Matthew) are trying to say, so I would suggest that you re-read it and make some changes to those 2 sections (2A and 2B). For chapter 3, it looks good so far, nicely explained but the tense used is "we" and it looks inconsistent because instead of "we", "the team" is being used in this document. So it has to be consistent. I will be re-reading everything again in the next update.

SVM

1.F	Chris Berkowitz	02/01/2013	Added chapter 3 D,E,F. Edits incoming.
1.G	Matthew Liang	02/01/2013	Rewrote Chapter 2 part A
1.H	Matthew Liang	02/01/2013	Add little changes to Chapter 2 part A. Fixed Chapter 2 part B
1.I	Chris Berkowitz	02/01/2013	Made "we" to "Team 5" changes.
1.J	Cruz Christopher	02/02/2013	Read through, change a lot of grammar, punctuation. Reworded sentences, removed sentences that made no sense.
1.K	Muhammad Naviwala	02/02/2013	Read through all the 3 chapters. Made some grammar changes and reworded sentences where necessary.
1.L	Christopher Cruz	02/05/2013	Final Review, Looks great.
1.M	Muhammad Naviwala	02/05/2013	Added the table of contents. Read through the 3 chapters. Looks good. Final review done.
1.N	Ryan Lee	02/05/2013	Team leader final review and approval for Red Deliverables.

TLs entry before RED TEAMS DELIVERABLES

Revision	Name	Due Date	Description
2.0	Lee, Ryan	02/05/2013	I changed Version to 2.0

DOCUMENT STORAGE

This file is stored in SVN at http://limi.cs.uh.edu/COSC4351/~~TEAM5~~/TEAM PROJECT DELIVERABLES/**Team Project Report.doc.**

team

Table of Contents:

Chapter 1: Introduction to TEAM5OAGS **5**
A. Introduction 5
B. Research Methodology 5
C. Report Organization 5

Chapter 2: Expanding the Requirements (Part 1: The Scope of Software Engineering) **6**
A. Problem Statement 6
B. Expanding the Borders 6
C. Expanding vs Replacing the System 7
D. Concluding Remarks 7

Chapter 3: Educating the User ("The Software Process" -Textbook & Problems with Software Production – BlackBoard) **7**
A. The Software Process 7
B. Problems with Software Complexity 8
C. Problems with Software Conformity 8
D. Problems with Software Changeability 8
E. Problems with Software Invisibility 8
F. Concluding Remarks 9

Report Format and Content

Chapter I: Introduction to TEAM5OAGS

A. Introduction

The purpose of this project is to construct a web application for an online art gallery. The company specializes in the selling of contemporary European and North American fine art such as lithographs, prints, paintings, and photographs. The web application that is to be constructed will allow customers to view and purchase artwork as well as services being offered by the company. The web application will also give merchants and former customers the opportunity to sell their artwork belongings to the company. By using standard software engineering principles and state of the art development tools, the goal of the team is to produce a high quality software product that will not only satisfy client's business needs, but also help improve its business operations.

B. Research Methodology

To determine the feasibility of the project to be undertaken, the team conducted a feasibility study and researched several websites that specialized in the selling of artwork. Examples of websites that were analyzed in the feasibility study were UGallery.com, artmajeur.com, redbubble.com, and 1-sharon-norman.artistwebsites.com. In the study, URLs for the websites and screenshots showing various user interface designs used by each site were compiled into a brief report and described in detail. Also, the web development framework, scripting languages, and servers that each website used were noted in the report. Finally, the team conducted a cost analysis for each website based on the data they have gathered and came up with the cost estimates for each website. Therefore, the team concludes that the project to be undertaken is indeed feasible and can be completed within the time and cost constraints that were estimated in the study.

C. Report Organization

The purpose of this project report is to document and describe the various software engineering principles, tools, and practices that will be utilized in the team project. The report will be divided into ten subsequent chapters where each chapter will focus on a different topic of software engineering. Each topic will be discussed in detail by organizing chapter content with problem statements, body paragraphs, and concluding remarks. The problem statements will present the background information of the topic to be discussed and what issues may need to be addressed. The body paragraphs will elaborate about the issues presented by the problem statement and seek to address them by providing examples and potential solutions. And finally, the concluding remarks section will summarize the topic that was covered by the chapter and may include any recommendations that the team has made for the project.

Chapter II: Expanding the Requirements (Part 1: The Scope of Software Engineering)

A. Problem Statement

History has presented software development as a challenging discipline with rare prospects of success. Although software engineers have been selective on applying theories, methods, and tools that are most appropriate, such practices have presented differences to the way software is developed. This often results in problems such as late delivery, being over budget, and software faults leading to unsuccessful (and unacceptable) outcomes. As TEAM5OAGS is developed, the team will apply the traditional software development practices and address issues from requirements to post-delivery maintenance in a technical and manageable fashion. From an economic perspective, software development comes with a high cost while significant progress has been made. In fact, most costs are associated with the maintenance of the product. For this reason, the team will be tasked with developing a product under a cost-efficient plan for both the company and the client. The product will undergo maintenance throughout the entire life-cycle to address corrections and improvements the software product may undergo.

B. Expanding the Borders

A software product can be modified after delivery to correct faults, enhance performance, or add new functionalities. Maintenance is an essential phase in the software life-cycle because the majority of the costs are associated with the maintenance of the product rather than its development. Also, if maintenance is not done properly, the inability to update software quickly and reliably will result in a lost business opportunity and early retirement for the software product.

Post-Maintenance consists of four different types. First, corrective maintenance involves removing faults or defects while leaving the specifications intact. Next, enhancement maintenance is a software update consisting of the changes to the specifications and the implementation. Third, perfective maintenance involves improvements to the software that add more functionality and efficiency beyond the initial requirements. Lastly, adaptive maintenance involves updating the software product in response to changes in the environment such as new hardware or new government regulations.

The project will undergo corrective maintenance in cases where correcting faults is a necessity. However, the system may also undergo perfective maintenance for improvement oat the request of the client. Overall, promoting a maintenance-friendly product will be the team's priority in terms of effort and spending so that the utility of the product to be developed can be enhanced.

C. Expanding vs. Replacing the System

While it is important to maintain the software, perfective and adaptive maintenances play a big role as they are both essential for allowing the software product to evolve

over time. During perfective maintenance, modifications to the software extend its functionality and improve its performance. This approach is carried out if a client wishes to add new features or modify it to run faster. Adaptive maintenance is performed when changes to the environment the system is running in occur. In such a case, it is imperative to update the system in order for it to fit in the new environment.

TEAM5OAGS will be addressed to handle such changes as the software is being developed. The maintenance will serve to update the system to adapt the new changes as well as surpass its initial requirements through enhancement. Maintenance is vital as it may influence and promote the extent to which this system was originally developed.

D. Concluding Remarks

It is important to recognize that software engineering practices are to be performed skillfully and insightfully. At every step of the software development cycle, they must be conducted in a balanced-budgeted way.

But it is also important to keep maintenance in mind when developing the product because there are bound to be faults in the software. Maintenance will help ensure that these faults are handled in a way that does not affect the integrity of the product. Eventually, the software product will grow to its potential as changes need to be made.

Chapter III: Educating the User ("The Software Process" -textbook & Problems with Software Production – BlackBoard)

A. The Software Process

Developing software is an intricate process that becomes much more manageable with a plan. Team 5 will be using the Unified Process methodology to ensure the software process will be structured, on schedule, and that the final product will perfectly match the product specified. The Unified Process was chosen as the methodology due to its wide acceptance as the standard object-oriented guideline and also because it is easily adaptable to properly fit any individual software needs.

The Unified Process has five key workflows. These workflows are requirements, analysis, design, implementation, and test workflows. Strictly adhering to these workflows will ensure that TEAM5OAGS will be effectively completed. The Unified Process requires modeling throughout the entire Software Development Life Cycle. After receiving the TEAM5OAGS requirements document, Team 5 began modeling via textual analysis. The team has properly analyzed the requirements and created UML use cases, UML class diagrams, and a Model-View Controller structure suitable for this web application. Through constant communication with the client, Team 5 refined the UML models to best capture the entire domain of the application that the client has envisioned.

B. Problems with Software Complexity

Designing software can spiral out of "complexity" control if allowed to do so. To combat this issue Team 5 has taken many steps to reduce the complexity as much as possible.

Each member of Team 5 completed the UML use cases and the UML class diagrams. Once complete, the document was refined from collaborated findings and created one perfect textual analysis that will help guide the development of this application towards the final deliverable. This will not only help lower the complexity of this process but will also keep the team on schedule for the checkpoints throughout the entire project. These models have also helped outline the exact specifications that TEAM5OAGS must have. Using these models and keeping communication constant amongst the team will greatly reduce the software complexity problem, and prevent other project hurdles such as scope creep and going over budget. Combining these processes and through continued use of the Unified Process, Team 5 will keep the software complexity issue at a minimum.

C. Problems with Software Conformity

Problems with software conformity are constantly arising. Parts of a larger software system may be upgraded and find they no longer work within the old product as intended. This is a serious issue that can ruin a project, destroy a necessary process, and even close an entire business down. This project has taken multiple steps to avoid this issue. For this reason, this is why Team 5 has decided to follow the Model-View Controller format. This means that Team 5 will not have to have the product conform to other preexisting products, where integration issues could arise. Team 5 will create all of the parts and the MVC form will help the team maintain this application's modularity for any future system upgrades.

D. Problems with Software Changeability

To address the problems with software changeability Team 5 has decided to implement a few simple principles that will be followed throughout the entire SDLP. Team 5 is going to, as previously mentioned, use the MVC format for TEAM5OAGS. This will greatly reduce many issues regarding ease of changeability in the future. By keeping the code in the MVC form the parts of the system will be much more accessible and therefore changeable if need be. TEAM5OAGS will also be heavily documented and the team will ensure that this documentation takes place throughout the entire process. This will help the users and any future system upgrade with the ability to succeed and understand thoroughly exactly what is going on within the application.

E. Problems with Software Invisibility

The problem with software invisibility lies within the actual software design itself. Team 5 has put considerable effort into creating TEAM5OAGS with the end user in mind. By keeping the application as modular as feasibly possible the application will

allow for extended use of the core functionality while also catering to the ever-changing technology frontier. This affects invisibility because when the application is written there is no emphasis on the end users having any ideas on how the actual processes interact with one another. Instead, they only understand how to use the application for its intended purpose. To combat the issue of software invisibility, Team 5 has made the effort to train (and continue training in the future) the end user as for a larger understanding of the back end that is typically unrealized. This will allow for future modification if need be and a firmer understanding of why the program acts the way it does. Team 5 has also documented every action taken towards the completion of TEAM5OAGS so that any third parties who may need to modify any parts of this application will be able to understand the parts separately and as a whole. This will keep future maintenance and upgrade costs at a much lower rate than the typical software products being produced are costing.

F. Concluding Remarks

The software process is a tedious process that leaves room for many problems throughout the entire cycle. Team 5 has taken an exuberant amount of precautions to deliver TEAM5OAGS without errors or flaws. With repeated communication with the client, Team 5 has gotten to understand exactly how TEAM5OAGS needs to function. Using the models and taking the necessary steps to refine the design and perfect the Model-View Controller methodology, Team 5 has hammered out a perfect model to follow during the coming design and implementations phases. TEAM5OAGS will not only work as the premier art gallery web application it is being designed for, but will also save the client on any future endeavors that the client may have in respect to the application.

Chapter IV: Choosing the Right Team ("Teams" - textbook)

A. Problem Statement
B. Assembling the Team

i. The Democratic Approach
ii. The Classical Approach (Chief Programmer)
iii. The Modern Approach (Chief Programmer & Business Manager)
iv. Additional Team Approaches

C. Recommendation for ***Team Project***
D. Concluding Remarks

Chapter V: Choosing the Right Model ("Software Life-Cycle Models" - textbook)

A. Problem Statement
B. Comparing Models

i. Waterfall Model
ii. Incremental Model
iii. Extreme Programming
iv. Synchronize and Stabilize
v. Spiral Model
vi. Object-Oriented Model

C. Recommendation for ***Team Project***
D. Concluding Remarks

Chapter VI: Analysis and Development Methods ("The Tools of Trade" - textbook)

A. Problem Statement
B. Conventional Engineering Methods

i. Stepwise Refinement
ii. Cost Benefit Analysis
iii. Software Metrics
iv. CASE
v. Software Versions
vi. Configuration Management
vii. Problem Statement
viii. Other Applicable Engineering Tools

C. Recommendation for ***Team Project***

D. Concluding Remarks

Chapter VII: Testing ("Testing" - textbook)

A. Problem Statement
B. Quality Assurance

Nonexecution-Based Testing
Execution-Based Testing

C. Recommendation for ***Team Project***
D. Concluding Remarks

Chapter VIII: Developing *Team Project* Page Master and Home Page ("Requirements Workflow" - textbook)

A. Problem Statement
B. ***Team Project*** **Page Master and Home Page**
C. Concluding Remarks

Chapter IX: OO Analysis Models ("OO Analysis Workflow" - textbook)

(Develop ***Team Project*** UML OO Analysis Models)

A. Problem Statement
B. The UML Diagrams
C. Concluding Remarks

Chapter X: OO Design Models ("OO Design Workflow" - textbook)

(Develop ***Team Project*** UML OO Detailed Design models)

A. Problem Statement
B. The UML Diagrams
C. Concluding Remarks

Chapter XI: Deployment Diagram ("More on UML" - textbook)

(Develop ***Team Project*** UML Package and Deployment models)

A. Problem Statement
B. The UML Diagrams
C. Concluding Remarks