

Warehaus Media

Final Project for SW Engineering Class CSC 648-848

Section 01 Summer 2017

Team One

Lindsey Anne - lhogg@mail.sfsu.edu

Josh Reed

Jasmine Mann

Jason Wong

Tyler Huang

Bo Li

Raymond Feng

Final Product Summary	5
Milestone Documents	6
Milestone 1	7
1. Executive Summary	9
2. Use Cases	10
Guest User	10
Merchant User	10
Patron User	10
Administrator	10
3. Data Definition	11
4. Functional Specifications	12
5. Non-Functional Specifications	13
6. Competitive Analysis	14
7. High-Level System Architecture	15
Web Application stack - LAMP	15
Deployment - Amazon Web Services (AWS)	15
Framework - CakePHP	15
UI - Bootstrap/JQuery/React	15
MVC Framework	15
8. Team Members	16
9. Checklist	17
Milestone 2	18
1. Executive Summary	20
2. Use Cases	21
Guest User	21
Merchant User	21
Patron User	21
Administrator	21
3. Data Definition	22
4. Functional Specifications	23
5. UI Mockups and Storyboards	24
5.1 Storyboard 1 - Guest (Unregistered) User	24

5.2 Storyboard 2 - Merchant (Unregistered) User	29
5.3 Storyboard 3 - Patron (Registered) User	32
6. Non-Functional Specifications	35
7. Competitive Analysis	36
8. High-Level System Architecture & Database Organization	37
8.1 High-Level System Architecture	37
Web Application stack - LAMP	37
Deployment - Amazon Web Services (AWS)	37
Framework - CakePHP	37
UI - Bootstrap/JQuery/React	37
MVC Framework	37
Media	37
Search	38
8.2 Database Tables	38
8.3 Entity-Relationship diagram	39
9. High-Level UML Diagrams	40
10. Key Risks	41
11. Team Organization	42
Milestone 3	43
Milestone Information	44
Checklist	44
Other things to check	44
Notes from Anthony	44
Milestone 4	45
1. Product Summary	47
2. Usability Test Plan	48
Test Objectives	48
Test Plan	48
Task: Search	48
Task: Details	48
Task: Upload	48
Test Plan Feedback	49
Questionnaire Form	49
3. QA Test Plan	50

Test Objectives	50
System Setup	50
Feature to be Tested	50
Test Cases	50
Test 1 - Searching by Text	50
Test 3 - Searching by Random Text	51
Test Results	51
4. Code Review	52
5. Self-Check on Best Practices for Security	54
6. Adherence to Non-Functional Specifications	55
Final Product Screenshots	56
Home Page	56
Registration Page	57
Login Page	58
Upload Page	59
Search Results	60
Photo Details	61
User Dashboard, a.k.a. "My Stuff"	62
Message Page	63
Database Schema	64

Final Product Summary

Warehaus Media is a site made by our team of 7 San Francisco State University students. Our site is a web-based service specializing in the buying and selling of users' stock images. A few of our functional specs are:

- Ability to browse site as a guest user
- Guest Users can register for an account using an email, username, and encrypted password
- Registered Users can upload media they want to sell
- Ability to search for media by title and category
- Users can message sellers to buy media
- In the backend, admin can manage the database, which includes being able to removing images and users

Our website can be accessed by visiting <http://sfsuse.com/~su17g01/warehausmedia>

Milestone Documents

The following pages are the final versions of Milestones 1-4.

Milestone 1

Initial Proposal of Warehaus Media

Team One

Lindsey Anne - lhogg@mail.sfsu.edu

Josh Reed

Jasmine Mann

Jason Wong

Tyler Huang

Bo Li

Raymond Feng

1. Executive Summary	3
2. Use Cases	4
Guest User	4
Merchant User	4
Patron User	4
Administrator	4
3. Data Definition	5
4. Functional Specifications	6
5. Non-Functional Specifications	7
6. Competitive Analysis	8
7. High-Level System Architecture	9
Web Application stack - LAMP	9
Deployment - Amazon Web Services (AWS)	9
Framework - CakePHP	9
UI - Bootstrap/JQuery/React	9
MVC Framework	9
8. Team Members	10

1. Executive Summary

People love art. Pictures, paintings, videos, or any other types of attractive visuals created and shared among people can evoke strong emotions and memories. According to hubspot.com, over 150 million daily active Snapchatters snap an estimate of 8,796 photos or memories per second. With that being said, people love the sight of a beautiful picture, because the image not only provides an immediate message to the receiver but capturing or collecting images is also a hobby for many of the people today.

In today's world of competitive trade and technology, businesses are aware that one of the most important focus is to constantly make sure that their marketing materials are visually appealing to their audience. With nice looking photos and awesome videos, customers will likely have a greater experience and stay. Media managers need a supply of images and videos to help their businesses better advertise and sell themselves.

This is where Warehaus Media comes in. From big companies to startups, Warehaus Media can provide users with visually attractive components to better attract users. The company can also provide better customer service than other popular media purchasing sites, such as iStock.

Warehouse Media is built by a group of well-trained engineers from San Francisco State University. The company's focus is to give users a great time by having a simple interface and an efficient searching and sorting method that will ensure the best experience when visiting.

2. Use Cases

Guest User

Kevin is an art student. He has never learned about design, and his professor assigns a project to him. He wants to look for some pictures or videos to buy as reference to inspire him to finish the project. As an unregistered guest user, he can only view the information (item's description, customers' reviews) of the item that he searched. Also, if some information shows illegal items, he can report directly to the website administrator. If he decides to buy an item, he is prompted to register an account.

Merchant User

Jonny is a self-employed graphic designer with a lot of photos and art that he would like to sell. He registers an account on the Warehaus Media website. After filling in his merchant information, he posts his items with appropriate prices and descriptions. Upon posting, all Guests and Patrons who search the specific items could see the information that he posted, and they can contact him directly to ask more details about the items.

Patron User

Steve is a frequent customer and needs another stock photo for his expanding tour guide service website. When Steve visits the Warehaus Media website, he signs into his account, where he is greeted and recommended cityscape pictures based on previous transactions and searches. Steve, going off the website's recommendations, finds another similar stock photo that he enjoys, so he proceeds to purchase. Since Steve is already registered and has saved his payment information so he can skip most of the Guest User's registration process and head straight to buying the product.

Administrator

Alex has been hired as the administrator of Warehaus Media. He will manage the database of the website and receive messages from registered and Guest users about illegal media files, complaints, and site bugs. He is easily able to delete media files which violate the Terms and Conditions of the site and send warnings to registered sellers. He is able to suspend/delete any registered accounts which violate the site's policies.

3. Data Definition

1. Registered user – user with an account that can either post or buy content
 - a. Email
 - b. Username
 - c. Password
2. Guest user – user without an account; can browse the site for content; Cannot post or buy content
3. Merchant – registered user selling media files
4. Patron - registered user buying media files
5. Content - The media files users will be selling--images or videos
 - a. Owner ID
 - b. Media Type
 - c. Category
 - d. Title
 - e. Description
 - f. Price
 - g. Thumbnail
 - h. Full-size media file
6. Administrator – a user with special privileges; will be able to delete any media files and ban any registered accounts; will be able to manage the database.

4. Functional Specifications

1. Guest users shall be able to browse through the site, but not have access to post or download media files.
2. Guest users shall be able to contact administrator.
3. Guest users shall not be able to contact sellers.
4. Guest users shall be able to create accounts and log in.
5. Merchant users shall be able to upload and post photos and videos they wish to sell.
6. Merchant users shall be able to edit or remove their listings.
7. Patron users shall be able to search for media files by owner, media type, and category.
8. Merchant and Patron users shall be able to reset their passwords and add/edit personal information as they wish.
9. Merchant and Patron users shall be able to find their passwords if they forgot, and passwords shall be sent to their email addresses.
10. Merchant and Patron users shall be able to contact other Merchants, Patrons, and Administrator.
11. Administrator shall be able to delete any listed media files.
12. Administrator shall be able to delete any registered accounts.
13. Administrator shall manage the database.
14. The site shall present the Terms and Conditions and Privacy Policy during the registration process.
15. The site shall ask for username and password.
16. Passwords shall be encrypted.
17. Fuzzy string search shall be used to filter photos and videos.
18. The site shall show most viewed media files if no file matches user's search.

5. Non-Functional Specifications

1. Application shall be developed using class provided LAMP stack
2. Application shall be developed using pre-approved set of SW development and collaborative tools provided in the class. Any other tools or frameworks must be explicitly approved by Anthony Souza on a case by case basis.
3. Application shall be hosted and deployed on Amazon Web Services as specified in the class
4. Application shall be optimized for standard desktop/laptop browsers, and must render correctly on the two latest versions of all major browsers: Mozilla, Safari, Chrome.
5. Application shall have responsive UI code so it can be adequately rendered on mobile devices but no mobile native app is to be developed
6. Data shall be stored in the MySQL database on the class server in the team's account
7. Application shall be deployed from the team's account on AWS
8. No more than 50 concurrent users shall be accessing the application at any time
9. Privacy of users shall be protected and all privacy policies will be appropriately communicated to the users.
10. The language used shall be English.
11. Application shall be very easy to use and intuitive. No prior training shall be required to use the website.
12. Google analytics shall be added
13. Messaging between users shall be done only by class approved methods and not via e-mail clients in order to avoid issues of security with e-mail services.
14. Pay functionality (how to pay for goods and services) shall not be implemented.
15. Site security: basic best practices shall be applied (as covered in the class)
16. Modern SE processes and practices shall be used as specified in the class, including collaborative and continuous SW development
17. The website shall prominently display the following text on all pages "*SFSU Software Engineering Project, Summer 2017. For Demonstration Only*". (Important so as to not confuse this with a real application).

6. Competitive Analysis

	Warehaus Media	Shutterstock	Depositphotos
Filter	+	+	+
Preview	+	+	+
Image	+	+	+
Video	+	+	+
Music	-	+	+
Simple, Clean UI	+	-	-

KEY:

- + Feature exists
- Feature does not exist

7. High-Level System Architecture

Web Application stack - LAMP

- Linux operating system
- Apache Web Server
- MySQL Database
- PHP Framework

Deployment - Amazon Web Services (AWS)

- Dependable cloud based web service
- Great scalability and speed

Framework - CakePHP

- Rapid development
- Active community support
- MVC architecture

UI - Bootstrap/JQuery/React

- Cross browser maintainability
- Modularized and encapsulated

MVC Framework

- Modular
- Scalable
- Lightweight and testable
- Maintainable

8. Team Members

Lindsey Anne	CEO (Team Lead)
Josh Reed	CTO (Tech Lead)
Jasmine Mann	Front End Engineer
Tyler Huang	Front End Engineer
Jason Wong	Database Administrator
Bo Li	Back End Engineer
Raymond Feng	Back End Engineer

9. Checklist

- Team decided on basic means of communications: DONE
- Team found a time slot to meet outside of the class: ON TRACK
- CTO chosen and working out well so far: DONE
- Github master chosen: DONE
- Team ready and able to use the chosen framework: ON TRACK
- Skills of each team member defined and known to all: ON TRACK
- Team lead ensured that all team members read the final M1 and agree/understand it before submission: DONE

Milestone 2

Warehaus Media: Requirements, Specs, Architecture, UI Mockups

Team One

Lindsey Anne - lhogg@mail.sfsu.edu

Josh Reed

Jasmine Mann

Jason Wong

Tyler Huang

Bo Li

Raymond Feng

1. Executive Summary	3
2. Use Cases	4
Guest User	4
Merchant User	4
Patron User	4
Administrator	4
3. Data Definition	5
4. Functional Specifications	6
5. UI Mockups and Storyboards	7
6. Non-Functional Specifications	11
7. Competitive Analysis	12
8. High-Level System Architecture & Database Organization	13
Web Application stack - LAMP	13
Deployment - Amazon Web Services (AWS)	13
Framework - CakePHP	13
UI - Bootstrap/JQuery/React	13
MVC Framework	13
Media	13
Search	13
9. High-Level UML Diagrams	16
10. Key Risks	17
11. Team Organization	18

1. Executive Summary

People love art. Pictures, paintings, videos, or any other types of attractive visuals created and shared among people can evoke strong emotions and memories. According to hubspot.com, over 150 million daily active Snapchatters snap an estimate of 8,796 photos or memories per second. With that being said, people love the sight of a beautiful picture, because the image not only provides an immediate message to the receiver but capturing or collecting images is also a hobby for many of the people today.

In today's world of competitive trade and technology, businesses are aware that one of the most important focus is to constantly make sure that their marketing materials are visually appealing to their audience. With nice looking photos and awesome videos, customers will likely have a greater experience and stay. Media managers need a supply of images and videos to help their businesses better advertise and sell themselves.

This is where Warehaus Media comes in. From big companies to startups, Warehaus Media can provide users with visually attractive components to better attract users. The company can also provide better customer service than other popular media purchasing sites, such as iStock.

Warehouse Media is built by a group of well-trained engineers from San Francisco State University. The company's focus is to give users a great time by having a simple interface and an efficient searching and sorting method that will ensure the best experience when visiting.

2. Use Cases

Guest User

Kevin is an art student. He has never learned about design, and his professor assigns a project to him. He wants to look for some pictures or videos to buy as reference to inspire him to finish the project. As an unregistered Guest user, he can only view the information (item's description, price and thumbnail) of the item that he searched. If he decides to buy an item, he is prompted to register an account.

Merchant User

Margaret is a self-employed graphic designer with a lot of photos and art that she would like to sell. She registers a Merchant account on the Warehaus Media website. After filling in her Merchant information, she posts her items with appropriate prices and descriptions. Upon posting, all Guests and Patrons who search the specific items could see the information that she posted, and they can contact her directly to ask more details about the items.

Patron User

Steve is a frequent customer and needs another stock photo for his expanding tour guide service website. When Steve visits the Warehaus Media website, he signs into his Patron account, where he is greeted and recommended cityscape pictures based on previous transactions and searches. Steve, going off the website's recommendations, finds another similar stock photo that he enjoys, so he proceeds to purchase. Since Steve is already registered and has saved his payment information so he can skip most of the Guest user's registration process and head straight to buying the product.

Administrator

Alexandra has been hired as the Administrator of Warehaus Media. She will manage the database of the website and receive messages from Merchant, Patron, and Guest users about illegal media files, complaints, and site bugs. She is easily able to delete media files which violate the Terms and Conditions of the site and send warnings to registered sellers. She is able to suspend/delete any registered accounts which violate the site's policies.

3. Data Definition

1. Registered user – user with an account that can either post or buy content depending on user type
 - a. Email
 - b. Username
 - c. Password
 - d. Role
2. Guest user – user without an account; can browse the site for content; Cannot post or buy content
3. Merchant – registered user with Merchant role; sells media files
4. Patron - registered user with Patron role; purchases media files
5. Content - The media files users will be selling and purchasing--images or videos
 - a. Owner ID
 - b. Media Type
 - c. Category
 - d. Title
 - e. Description
 - f. Price
 - g. Thumbnail
 - h. Full-size media file
6. Administrator – a user with special privileges; will be able to delete any media files and ban any registered accounts; will be able to manage the database.

4. Functional Specifications

1. Guest users shall be able to browse through the site, but not have access to post or download media files. - P1
2. Guest users shall be able to create accounts and log in. - P1
3. Merchant users shall be able to upload and post photos and videos they wish to sell. - P1
4. Patron users shall be able to search for media files by name, owner, media type, and category. -P1
5. Patron users shall be able to have access to download media files. - P1
6. Administrator shall manage the database. - P1
7. The site shall present the Terms and Conditions and Privacy Policy during the registration process. - P1
8. The site shall ask for username and password. - P1
9. Passwords shall be encrypted. - P1
10. When creating accounts, the user shall fill in user name, double passwords, (one for create, one for double check), email address (in order to find passwords if they forgot). - P1

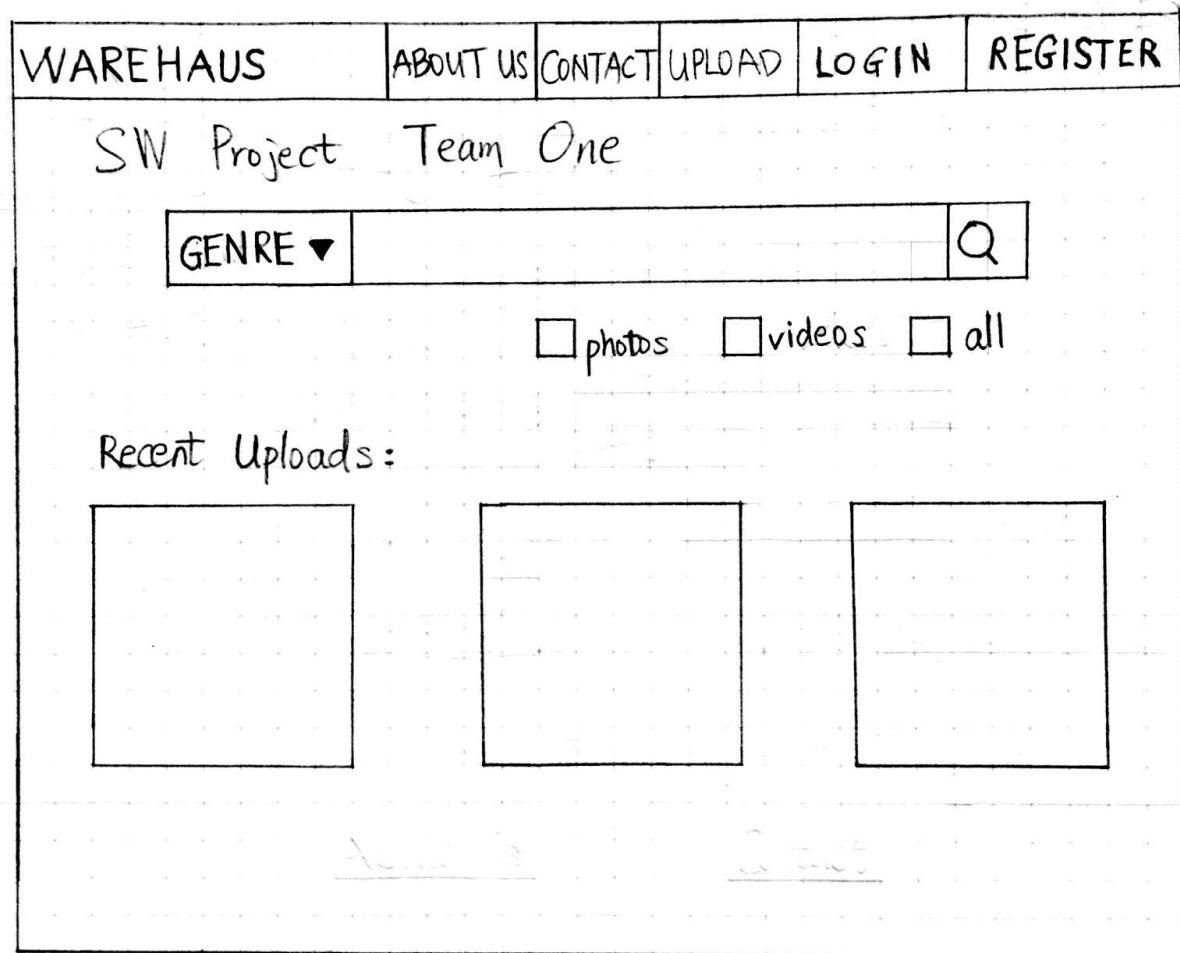
11. Guest users shall be able to contact administrator. - P2
12. Guest users shall not be able to contact sellers. - P2
13. Merchant users shall be able to edit or remove their listings. - P2
14. Merchant and Patron users shall be able to reset their passwords and add/edit personal information as they wish. - P2
15. Merchant and Patron users shall be able to find their passwords if they forgot, and passwords shall be sent to the their email addresses. -P2
16. Merchant and Patron users shall be able to contact other Merchants, Patrons, and Administrator. - P2
17. Administrator shall be able to delete any listed media files. -P2
18. Administrator user can be activated/deactivated in order to manage Administrative permissions. -P2
19. Fuzzy string search shall be used to filter photos and videos. - P2

20. Administrator shall be able to delete any registered accounts. -P3
21. The site shall show most viewed media files if no file matches user's search. - P3
22. The site shall have the order lists which are history and current orders. - P3
23. The site shall have shopping cart which photos and videos are selected by Patron users without purchasing. - P3

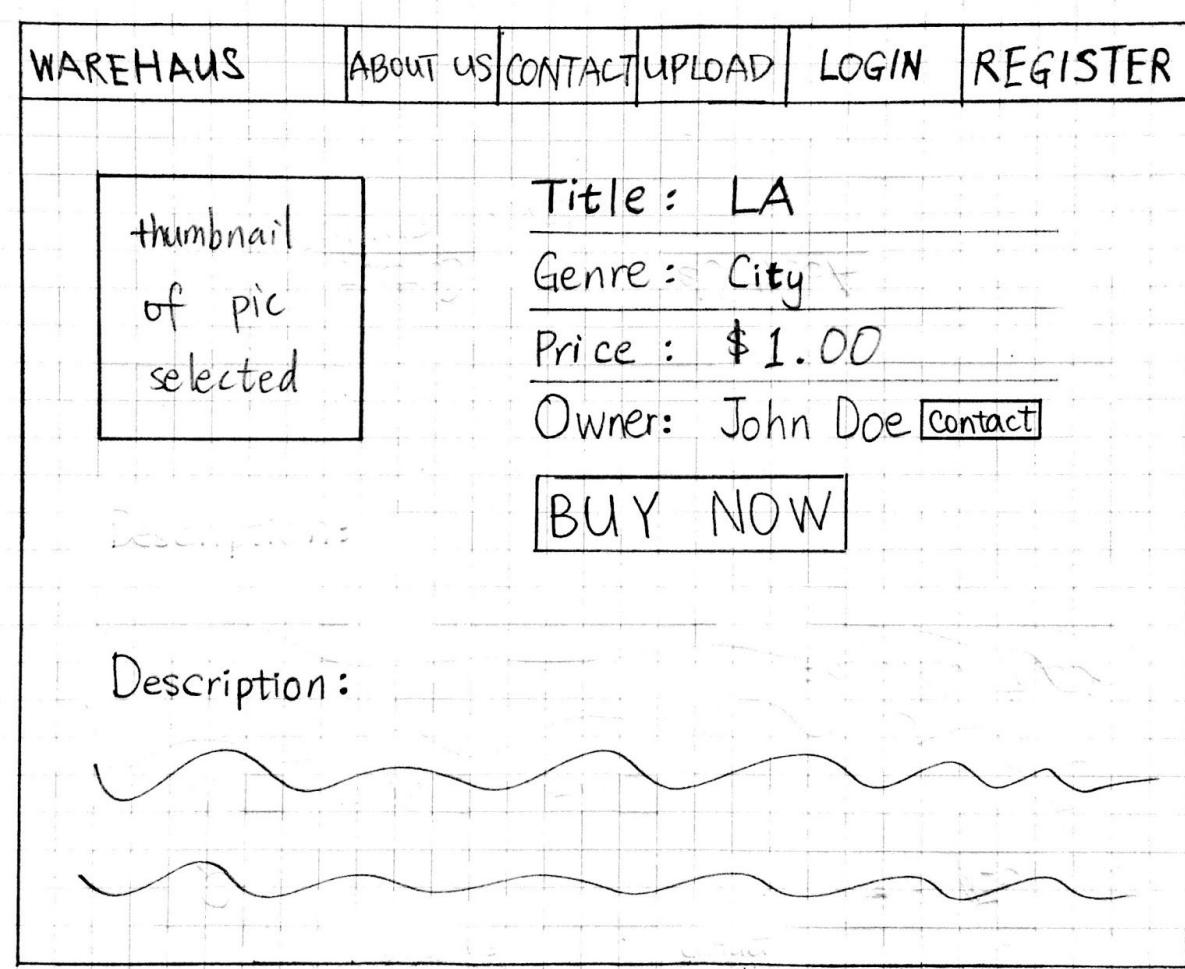
5. UI Mockups and Storyboards

5.1 Storyboard 1 - Guest (Unregistered) User

Kevin enters the homepage of Warehaus Media where he can see there is a search bar where he can search media by keyword. Also, the three most recent pictures will be displayed under that search bar.



Kevin clicks on a picture shown on the homepage which he is interested in, which takes him to the detailed information about the picture. Name, category, price and the owner of the picture will be displayed. Kevin can click on "Contact" next to the owner name in order to contact the owner John Doe. He may click "Buy Now" if he decides to purchase the picture.



Kevin wants to purchase the picture, so he clicks on "Buy Now". Because he is an unregistered user, he is sent to the login page. If Kevin has an account, he can provide his username and password in order to login; if Kevin does not have an account yet, he can either click the "Register" button on the upper right corner or click the hyperlink "Register Here," which will lead him to the registration page.

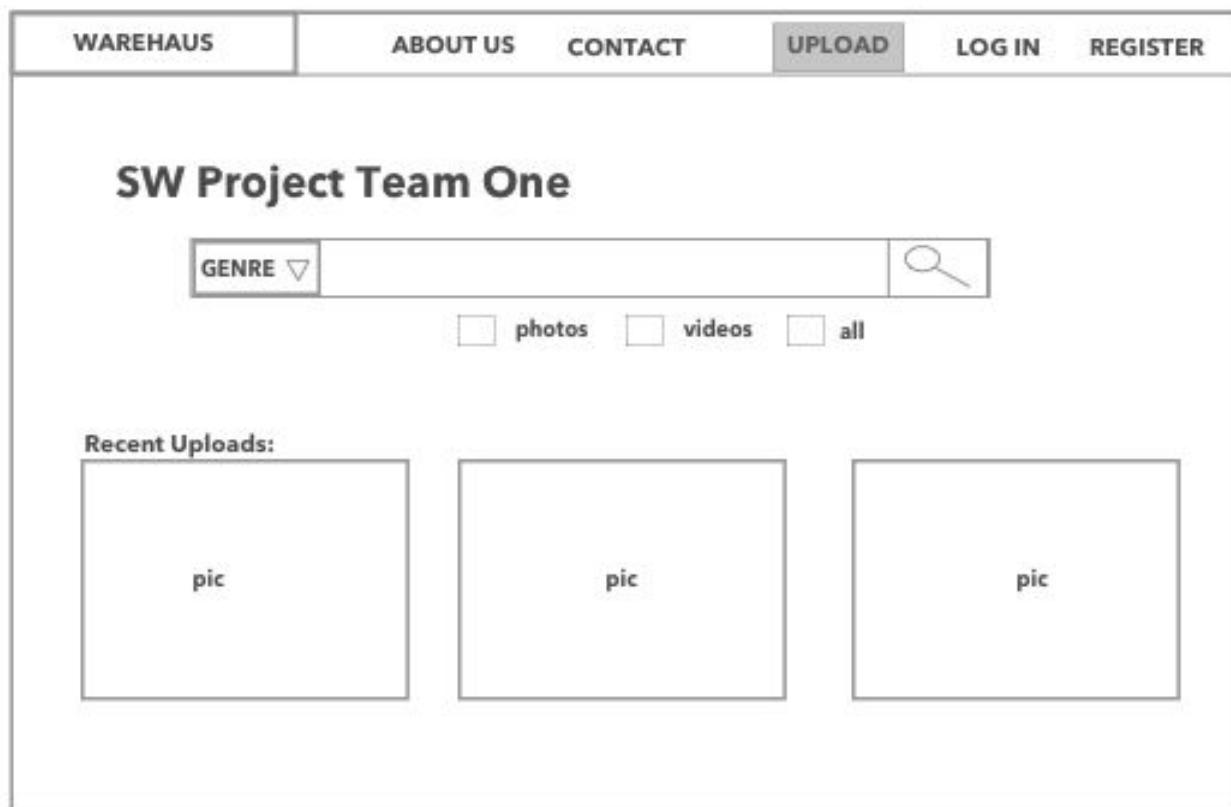
WARE HAUS	ABOUT US	CONTACT	UPLOAD	LOGIN	REGISTER
<p>Login:</p> <p>Username: <input type="text"/></p> <p>Password : <input type="password"/></p> <p><input type="button" value="Login"/></p> <p>Don't have an account? <u>Register here</u></p> <p><u>Forgot password</u></p>					

If Kevin does not have an account and needs to register, this is the page he will see, he will be asked to provide username and password, and he needs to confirm the password he picks again. Also, he needs to agree with the Terms and Conditions of the Warehaus Media in order to continue.

WAREHAUS	ABOUT US	CONTACT	UPLOAD	LOGIN	REGISTER
<p>Don't have an account ? Register is easy !</p> <p>Username : <input type="text"/></p> <p>Password : <input type="text"/></p> <p>Confirm Password : <input type="text"/></p> <p><input type="checkbox"/> I agree with <u>Terms and Conditions</u></p> <p><input type="button" value="Register"/></p> <p>Have an account already? <u>Login here</u></p>					

5.2 Storyboard 2 - Merchant (Unregistered) User

Margaret has a collection of pictures she would like to sell. She goes to Warehaus and clicks the upload button at the top of the page.



Because Margaret does not have an account, she is taken to the registration page where she fills out her email, username, and password, and she clicks the checkbox to agree to the terms and conditions. She then clicks "Submit."

Register

Email:

Username:

Password:

Confirm Password:

I agree with Terms and Conditions.

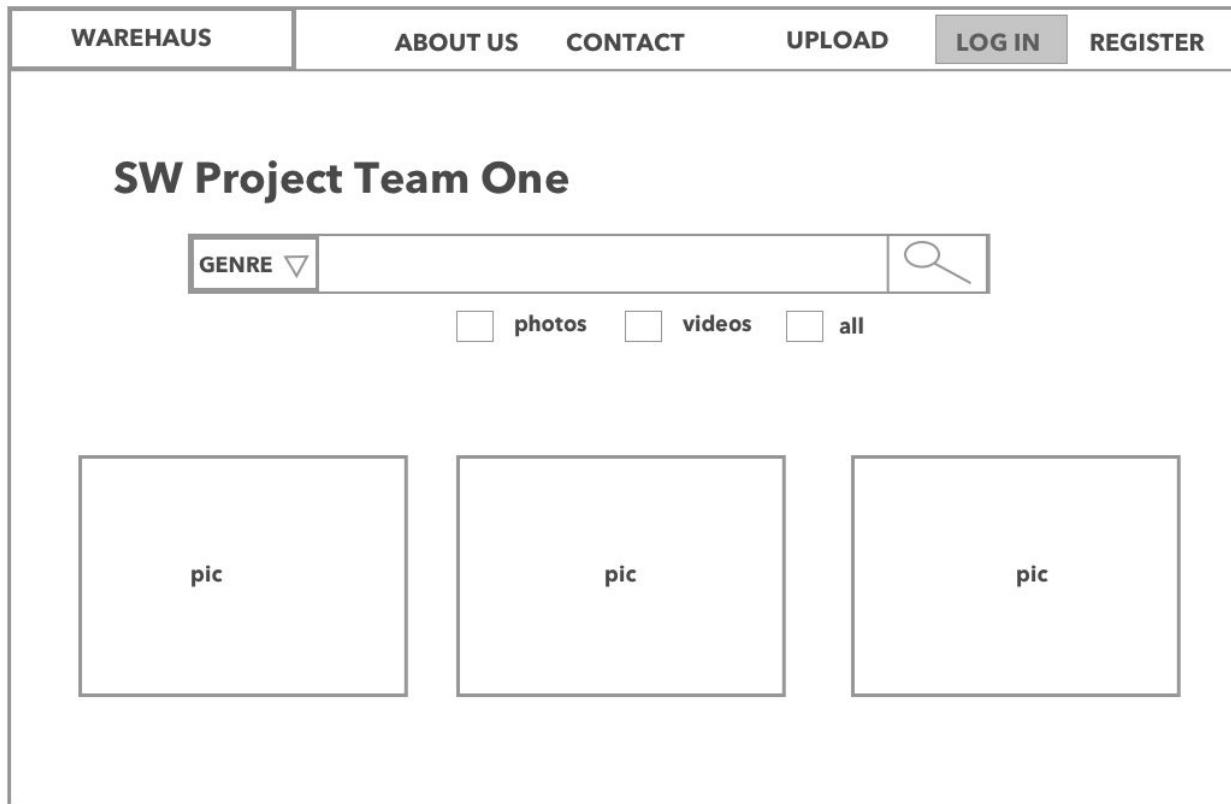
Have an account already? [Log-in Here!](#)

Margaret then arrives at the upload page where she can upload her media.

WAREHAUS	ABOUT US	CONTACT	UPLOAD	MANAGE UPLOADS	SIGN OUT
<p style="text-align: center;">click here to upload</p>					
CATEGORY ▼		SET PRICE		\$0.00	
<p>TITLE</p> <p>DESCRIPTION</p>					
SUBMIT					

5.3 Storyboard 3 - Patron (Registered) User

Steve is a frequent customer. When he visits the website, he clicks the login link.



The storyboard shows a user interface for a website. At the top, there is a navigation bar with links: WAREHAUS, ABOUT US, CONTACT, UPLOAD, LOG IN (which is highlighted in grey), and REGISTER. Below the navigation bar, the title "SW Project Team One" is displayed in a large, bold font. Underneath the title is a search bar with a dropdown menu labeled "GENRE" and a magnifying glass icon. Below the search bar are three checkboxes labeled "photos", "videos", and "all". Below the search area are three placeholder boxes, each containing the text "pic".

From here he enters his information and clicks submit

WAREHAUS	ABOUT US	CONTACT	UPLOAD	LOG IN	REGISTER
<p style="text-align: center;">Log in</p> <p>Username: <input type="text"/></p> <p>Password: <input type="password"/></p> <p style="text-align: center;">Forgot your password? click here</p> <p style="text-align: center;"><input type="button" value="Submit"/></p>					

Once logged in, the homepage has been personalized for Steve. He can go off of the website's suggestions to find an image similar to what he already likes.

SW Project Team One

GENRE ▼

 photos videos all**suggested for you:**

pic

pic

pic

6. Non-Functional Specifications

1. Application shall be developed using class provided LAMP stack
2. Application shall be developed using pre-approved set of SW development and collaborative tools provided in the class. Any other tools or frameworks must be explicitly approved by Anthony Souza on a case by case basis.
3. Application shall be hosted and deployed on Amazon Web Services as specified in the class
4. Application shall be optimized for standard desktop/laptop browsers, and must render correctly on the two latest versions of all major browsers: Mozilla, Safari, Chrome.
5. Application shall have responsive UI code so it can be adequately rendered on mobile devices but no mobile native app is to be developed
6. Data shall be stored in the MySQL database on the class server in the team's account
7. Application shall be deployed from the team's account on AWS
8. No more than 50 concurrent users shall be accessing the application at any time
9. Privacy of users shall be protected and all privacy policies will be appropriately communicated to the users.
10. The language used shall be English.
11. Application shall be very easy to use and intuitive. No prior training shall be required to use the website.
12. Google analytics shall be added.
13. Messaging between users shall be done only by class approved methods and not via e-mail clients in order to avoid issues of security with e-mail services.
14. Pay functionality (how to pay for goods and services) shall not be implemented.
15. Site security: basic best practices shall be applied (as covered in the class)
16. Modern SE processes and practices shall be used as specified in the class, including collaborative and continuous SW development
17. The website shall prominently display the following text on all pages "*SFSU Software Engineering Project, Summer 2017. For Demonstration Only*". (Important so as to not confuse this with a real application).
18. Support all current versions of Chrome, Firefox, Edge, and Internet Explorer. Older versions shall not be explicitly supported.

7. Competitive Analysis

	Warehaus Media	Shutterstock	Depositphotos
Filter	+	+	+
Preview	+	+	+
Image	+	+	+
Video	+	+	+
Music	-	+	+
Simple, Clean UI	+	-	-

KEY:

- + Feature exists
- Feature does not exist

8. High-Level System Architecture & Database Organization

8.1 High-Level System Architecture

Web Application stack - LAMP

- Linux operating system
- Apache Web Server
- MySQL Database
- PHP Framework

Deployment - Amazon Web Services (AWS)

- Dependable cloud based web service
- Great scalability and speed

Framework - CakePHP

- Rapid development
- Active community support
- MVC architecture

UI - Bootstrap/JQuery/React

- Cross browser maintainability
- Modularized and encapsulated

MVC Framework

- Modular
- Scalable
- Lightweight and testable
- Maintainable

Media

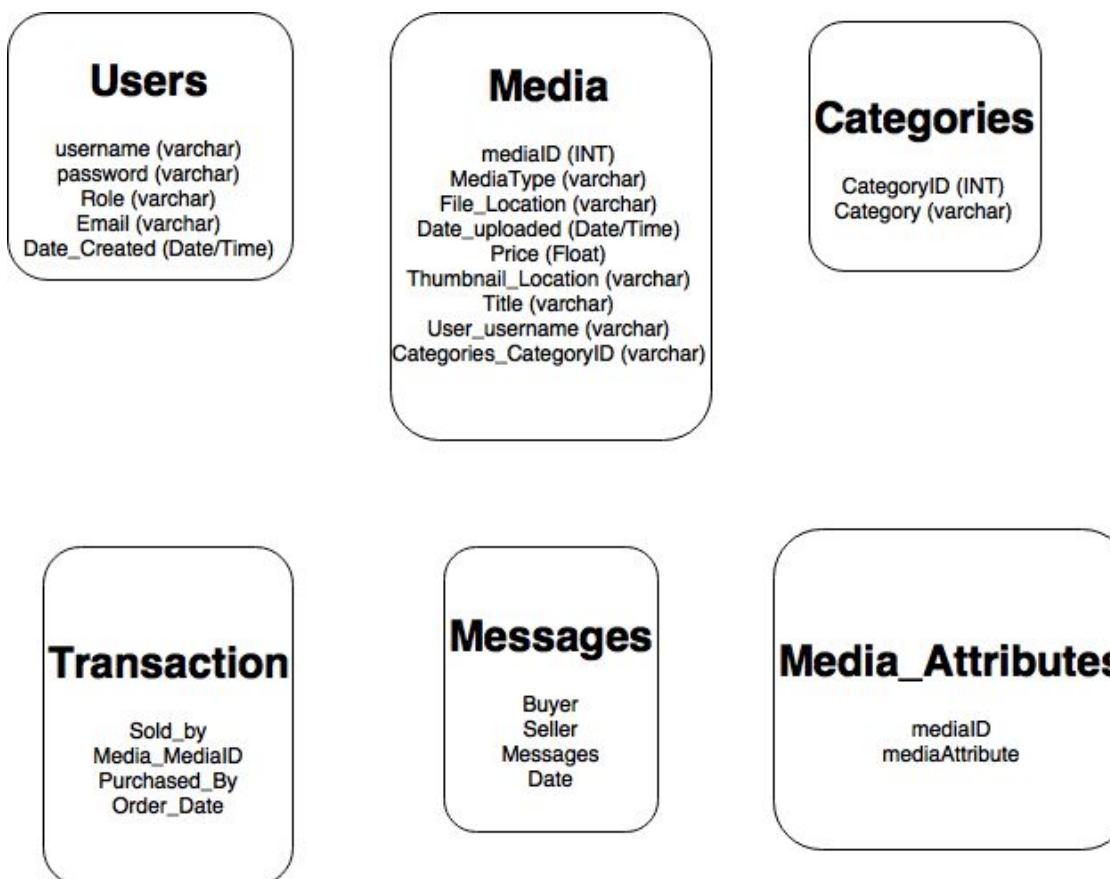
- Image supported formats are JPG,PNG,BMP,BPG
- Video supported formats are AVI,FLV,MP4,MOV
- Images will be accepted in 1280x720 or 1920x1080 resolution
- Videos must have a 720x480 resolution
- Videos and images will be kept within an organized file system

Search

- Search bar will include a drop down menu with category and genre to expedite the searching process
- Search bar will use SQL built in LIKE, BETWEEN, and IN to return fast and correct results

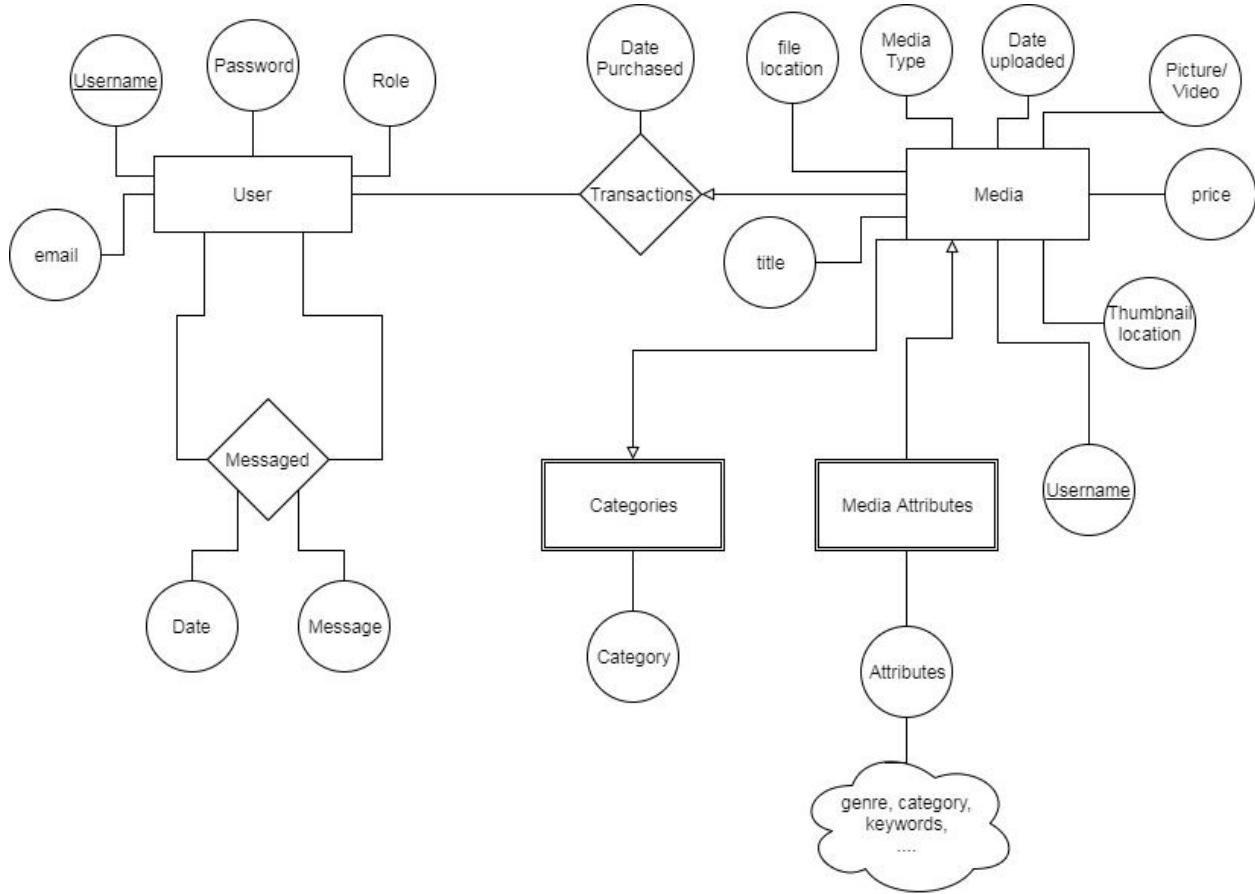
8.2 Database Tables

The database will be comprised of 6 tables: Users, Media, Categories, Transaction, Messages, and Media_Attributes.

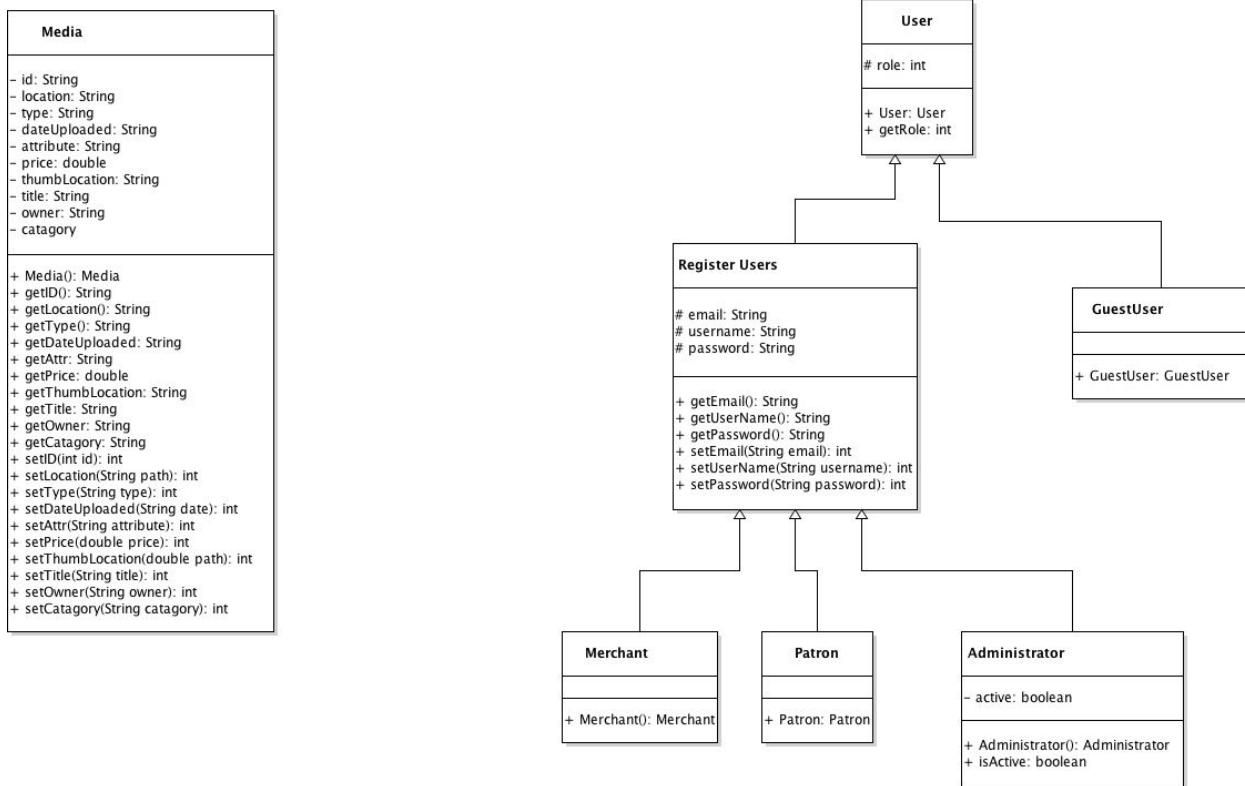


8.3 Entity-Relationship diagram

This ER relationship diagram depicts how these tables will interact with other within the database:



9. High-Level UML Diagrams



10. Key Risks

1. Skills Risk

The team members' skill sets and levels vary greatly, which can be a disadvantage for those who are novices in some areas where a more advanced skill set is necessary. Knowledge transfer will be necessary from more experienced team members to the others, which will be done through various means such as workshops, tutorial sharing, and great communication.

2. Scheduling Risk

Our team members all have different time frames that they are available due to work or other outside activities. Due to conflicting time schedules it can be considerably hard to meet up during the week to go over what needs to be done with the project. One way that we can correct this issue is to agree to conference calls in the late afternoon when everyone is home. Another way to mitigate is to reduce the scope of the project.

3. Technical Risk

Our individual team members excel at most aspects of the project but it seems that we are currently lacking expertise in the database department. This can cause issues and stalling with the other parts of the project. To correct this problem each team member can work together to understand concepts more deeply and not be afraid to ask the CTO for help.

4. Teamwork Risk

Our individual team members all possess abundant skills. However; sometimes we lack the skill of good communication. This can create issues such as hinderance with branch and pull requests, misunderstanding of the assignments, and frustration. To fix this we can agree to more conference calls, join an online organization website such as trello to monitor activities, and communicate more with the CEO and CTO.

5. Legal/Content Risk

Using photos and videos downloaded from the internet carries a legal risk if any of the team members don't check that the media files are labeled for reuse. To mitigate this risk, the team will only use photos and videos created by fellow team members.

11. Team Organization

Lindsey Anne	CEO (Team Lead)
Josh Reed	CTO (Tech Lead)
Jasmine Mann	Front End Engineer
Tyler Huang	Front End Engineer
Jason Wong	Database Administrator
Bo Li	Back End Engineer
Raymond Feng	Back End Engineer

Milestone 3

Checklist for Instructors and Teams

Team One

Lindsey Anne - lhogg@mail.sfsu.edu

Josh Reed

Jasmine Mann

Jason Wong

Tyler Huang

Bo Li

Raymond Feng

Milestone Information

- Date: 7/19/17
- Team: 01
- Number of team members present: 5
- Instructor: Anthony Souza

Checklist

- Runs from deployment server
- Github organization (submission stats, organization of branches)
- Github usage: Comments on posting; Number of posting to github; Appr. even distribution of submissions among team members (check github post stats for all members)
- Code documented (header, in code) with good coding style
- MVC/OO patterns followed up
- Frameworks deployed correctly
- Database organization (good DB tables as we spoke, categories in separate table)
- Check search code
- Adherence to best practices of security (PW encrypted, inputs verified, (HTTPS not required since we do not have sec. cert for now))
- Other

Other things to check

- P1 set of features committed? Team is focused only on P1s?
- Teamwork: is the team working out, any related issues
- Digital content: status?
- Risks: all actual
- Comments and checkpoints (if any)

Notes from Anthony

Only front end views were shown, was not hooked up to backend

Some work also needs to be done in the backend (Controllers)

UI needed some structure organization and CSS

Code needs to be running from the class server and not laptops.

Database table creation needs to be done through cake

A checkpoint was set for end of the day, Saturday 7/29/2017:

The purpose of this checkpoint was for reworking how cake built the database tables, etc.

Milestone 4

Warehaus Media: Beta Release

Team One

Lindsey Anne - lhogg@mail.sfsu.edu

Josh Reed

Jasmine Mann

Jason Wong

Tyler Huang

Bo Li

Raymond Feng

1. Product Summary	3
2. Usability Test Plan	4
Test Objectives	4
Test Plan	4
Task: Search	4
Task: Details	4
Task: Upload	4
Test Plan Feedback	5
Questionnaire Form	5
3. QA Test Plan	6
Test Objectives	6
System Setup	6
Feature to be Tested	6
Test Cases	6
Test 1 - Searching by Text	6
Test 3 - Searching by Random Text	7
Test Results	7
4. Code Review	8
5. Self-Check on Best Practices for Security	10
6. Adherence to Non-Functional Specifications	11

1. Product Summary

Warehaus Media is a site made by our team of 7 San Francisco State University students. Our site is a web-based service specializing in the buying and selling of users' stock images. A few of our functional specs are:

- Ability to browse site as a guest user
- Guest Users can register for an account using an email, username, and encrypted password
- Registered Users can upload media they want to sell
- Ability to search for media by title and category
- Users can message sellers to buy media
- In the backend, admin can manage the database, which includes being able to removing images and users

Our website can be accessed by visiting <http://sfsuse.com/~su17g01/warehausmedia>

2. Usability Test Plan

Test Objectives

Assign tasks to test users to assess on user experience:

- Customers browse/search for images
- Customers send messages to merchants (owner of images)
- Merchants upload images
- Users manage their account using the dashboard

Test Plan

Task: Search

- Role: Customer.
- Open your browser and navigate to
<http://sfsuse.com/~su17g01/warehausmedia>
- Search for images in the Food category.
- Verify that images of food are displayed.
- Fill out the chart on the next page with feedback.

Task: Details

- Role: Customer.
- Open your browser and navigate to
<http://sfsuse.com/~su17g01/warehausmedia>
- Search for images in the Food category.
- Select an image and verify that its details are displayed.
- Fill out the chart on the next page with feedback.

Task: Upload

- Role: Merchant.
- Open your browser and navigate to
<http://sfsuse.com/~su17g01/warehausmedia>
- Register for an account, and login.
- Upload a sample photo using the Upload link and entering its details.
- Verify that the photo uploads to the website.
- Fill out the chart on the next page with feedback.

Test Plan Feedback

Task Name	% completed	Errors	Comments
Search			
Details			
Upload			

Questionnaire Form

Please answer the following questions with regards to the usability of the Warehaus Media website. Circle the response that you feel is most correct.

1. I thought the Warehaus Media website was easy to use.

Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
-------------------	----------	---------	-------	----------------

2. I thought there was too much inconsistency with the Warehaus Media website.

Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
-------------------	----------	---------	-------	----------------

3. I think that I would like to use the Warehaus Media website frequently.

Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
-------------------	----------	---------	-------	----------------

3. QA Test Plan

Test Objectives

The objective of the QA test is to thoroughly test the search function of Warehaus Media in order to ensure that search results display pictures relevant to the search query. Searching by text and category will be tested.

System Setup

Processor: 2.2 GHz Intel Core i7

Memory: 16 GB 1600 MHz DDR3

Operating System: macOS Sierra Version 10.12.6

Browser 1: Safari Version 10.1.2 (12603.3.8)

Browser 2: Google Chrome Version 60.0.3112.90 (Official Build) (64-bit)

Feature to be Tested

This QA test will test the functionality of the search function from Warehaus Media's home page. The test will test for the following expected functionality:

All pictures which match the search query (text, category, or both) shall be shown.

Irrelevant pictures shall be shown if no picture match the search query.

All results should be readable and openable.

The search query shall not disappear from the search box after searching.

All results shall be the same across both browsers.

Test Cases

Test 1 - Searching by Text

Search for “food” on home page. Validate that every expected functionality is met. If all of the expected functionality is met, the test result shall be PASS. Perform the search on both browsers.

Test 2 - Searching by Category

Search for “architecture” on home page. Validate that every expected functionality is met. If all of the expected functionality is met, the test result shall be PASS. Perform the search on both browsers.

Test 3 - Searching by Random Text

Search for random text “segdfaergfsd” on home page. Validate that every expected functionality is met. If all of the expected functionality is met, the test result shall be PASS. Perform the search on both browsers.

Test Results

Test No.	Test Title	Test Description	Test Input	Expected Output	Test Result (Browser 1)	Test Result (Browser 2)
1	Searching by Text	Search for “food” by typing in search box	“food”	All pictures which contain “food” in the title	N/A	N/A
2	Searching by Category	Search for “architecture” by choosing it on drop-down menu	“architecture”	All pictures in the category of architecture	N/A	N/A
3	Searching by Random Text	Search for random text “akjhadf” by typing in search box	“akjhadf”	Some pictures	N/A	N/A

4. Code Review

Coding Style: To ensure that we make 100% use of the framework, we first made sure that we are all following all the naming conventions from tables names to classes and column names for posting and setting to and from the view. Second, we made sure that we use clear and meaningful variable names as well as method controller names so anyone will know what the view is going to do.

Code Review: We enforce strict review of everyone's code in order to make sure that software with a lot of obvious problems does not get into our main production branch. This process is done through Github. Once a developer is done with his/her part, the developer then pushes to a new branch on GitHub and creates a pull request with detailed comments and title specifying what he/she has done on the branch. Before merging is allowed, both the team lead and the tech lead will run his/her branch to ensure that the required functionality is functioning as requested. And if not, the reviewer can add comments on specific lines of code and request for changes.

The following page includes a sample of a pull request with code review remarks.

This repository Search Pull requests Issues Marketplace Gist

Unwatch 6 Star 0 Fork 0

Code Issues 6 Pull requests 4 Projects 0 Wiki Settings Insights

Mystuff #35

Open jhuang28 wants to merge 6 commits into master from mystuff

Conversation 6 Commits 6 Files changed 6 +231 -0

jhuang28 commented 3 days ago

@runongirlrunon "MY STUFF" button will appear when user logs in, and two buttons on Mystuff page. However, there's no Pictures page yet.

jhuang28 added some commits 3 days ago

- updated default.ctp and added mystuff.ctp b717434
- updated mystuff.ctp 3ceb5c2

runongirlrunon requested changes 2 days ago

Great start @jhuang28! Let's keep improving it. 👍

src/Template/Pages/mystuff.ctp

```

48 +      <h1>CSC 648 | Team One</h1>
49 +    </div>
50 +
51 +      <h2>SFSU Software Engineering Project, Summer 2017. For Demonstra

```

runongirlrunon 2 days ago Owner

You can probably take this div out, we have it on the main page and I think that's all we really need.

Reply...

src/Template/Pages/mystuff.ctp

```

54 +
55 +<body>
56 +<div class="content">
57 +  <h3>My stuff:</h3>

```

runongirlrunon 2 days ago edited Owner

You could edit this heading to say "Hello, <person's name here>!" with a simple PHP script to look at whoever is logged in.

Reply...

Reviewers runongirlrunon

Assignees No one—assign yourself

Labels None yet

Projects None yet

Milestone No milestone

Notifications

Unsubscribe You're receiving notifications because you were mentioned.

3 participants

Lock conversation

5. Self-Check on Best Practices for Security

Assets

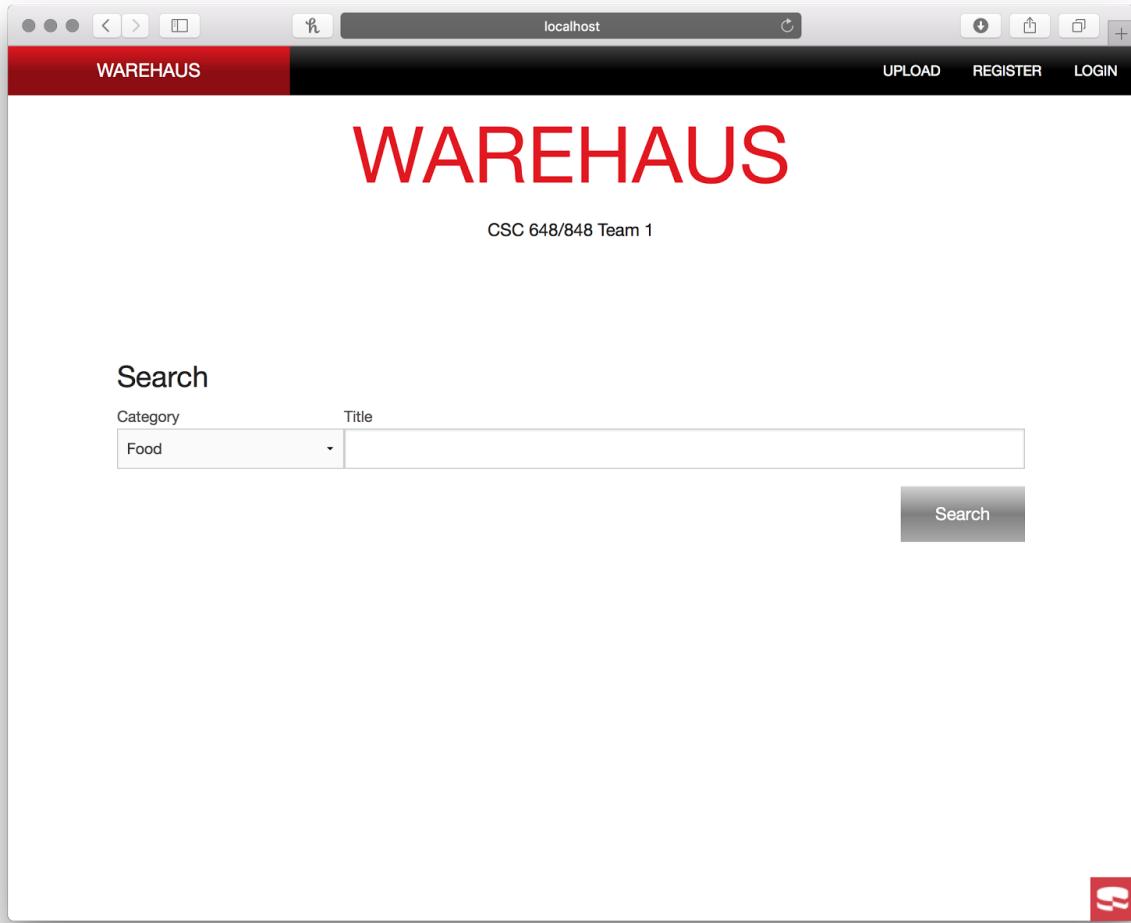
1. Data in database is hidden from users and can only be manipulated through controller interaction or administrative privilege.
2. Web application will be used entirely in a HTTPS environment to encrypt sensitive data such as passwords and credit card numbers
3. Images are securely kept on a filesystem within the Amazon Web Service and cannot be accessed remotely.
4. Users passwords are encrypted with bcrypt password hashing technology that uses a salt to protect against rainbow table attacks as well as brute-force search attacks.
5. Input is correctly validated to protect against SQL injection and database error due to malformed queries.

6. Adherence to Non-Functional Specifications

1. Application shall be developed using class provided LAMP stack (**Done**)
2. Application shall be developed using pre-approved set of SW development and collaborative tools provided in the class. Any other tools or frameworks must be explicitly approved by Anthony Souza on a case by case basis. (**Done**)
3. Application shall be hosted and deployed on Amazon Web Services as specified in the class (**Done**)
4. Application shall be optimized for standard desktop/laptop browsers, and must render correctly on the two latest versions of all major browsers: Mozilla, Safari, Chrome. (**On Track**)
5. Application shall have responsive UI code so it can be adequately rendered on mobile devices but no mobile native app is to be developed (**Done**)
6. Data shall be stored in the MySQL database on the class server in the team's account (**On Track**)
7. Application shall be deployed from the team's account on AWS (**Done**)
8. No more than 50 concurrent users shall be accessing the application at any time (**Done**)
9. Privacy of users shall be protected and all privacy policies will be appropriately communicated to the users. (**Done**)
10. The language used shall be English. (**Done**)
11. Application shall be very easy to use and intuitive. No prior training shall be required to use the website. (**Done**)
12. Google analytics shall be added. (**On Track**)
13. Messaging between users shall be done only by class approved methods and not via e-mail clients in order to avoid issues of security with e-mail services. (**Done**)
14. Pay functionality (how to pay for goods and services) shall not be implemented. (**Done**)
15. Site security: basic best practices shall be applied (as covered in the class) (**Done**)
16. Modern SE processes and practices shall be used as specified in the class, including collaborative and continuous SW development (**Done**)
17. The website shall prominently display the following text on all pages "*SFSU Software Engineering Project, Summer 2017. For Demonstration Only*". (Important so as to not confuse this with a real application). (**On Track**)
18. Support all current versions of Chrome, Firefox, Edge, and Internet Explorer. Older versions shall not be explicitly supported. (**Done**)

Final Product Screenshots

Home Page



Registration Page

Note: The form is not centered, which is a bug we intend to fix.

The screenshot shows a web browser window with a red header bar containing the text "WAREHAUS". The main content area has a title "Add User" followed by four input fields: "Username *", "Password *", "Role", and "Email". A "SUBMIT" button is located below the input fields. The browser's address bar shows "localhost". The top right of the browser window includes standard controls for zooming and navigating.

WAREHAUS

localhost

UPLOAD REGISTER LOGIN

Add User

Username *

Password *

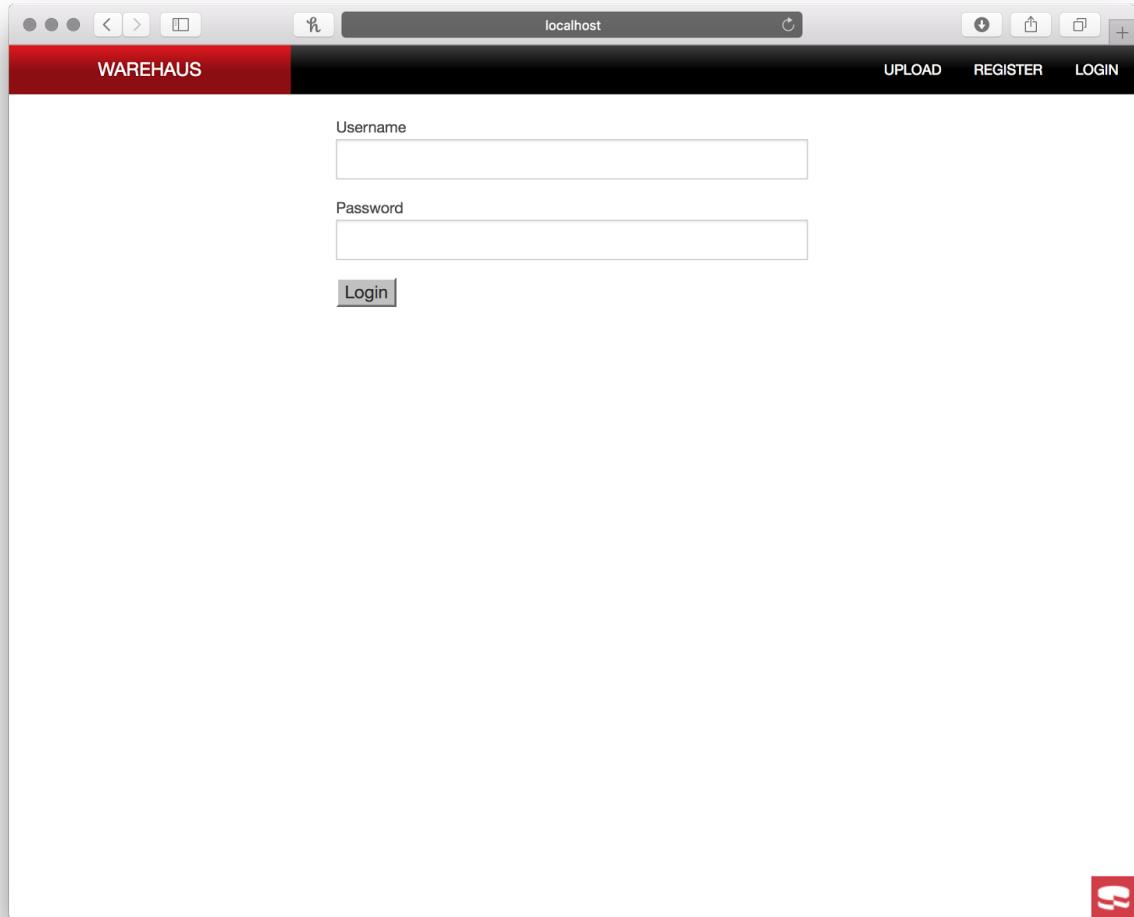
Role

Email

SUBMIT



Login Page



Upload Page

The screenshot shows a web browser window with the URL "localhost" in the address bar. The page has a red header bar with the word "WAREHAUS" in white. To the right of the header are three buttons: "UPLOAD", "MY STUFF", and "LOGOUT". Below the header, the main content area has a title "Add Media". The form consists of several input fields:

- Title:** An empty text input field.
- Description:** A large empty text area.
- Category:** A dropdown menu set to "Food".
- Price:** An empty text input field.
- Upload:** A file input field labeled "Choose File" with the message "no file selected".
- Create Product:** A button labeled "Create Product".

In the bottom right corner of the browser window, there is a small red square icon containing a white letter "S".

Search Results

localhost

WAREHAUS UPLOAD MY STUFF LOGOUT

Search Results



Tacos

Details



Coffee

Details



Photo Details

localhost

WAREHAUS

UPLOAD MY STUFF LOGOUT



Tacos

Category: Food

Price: \$2

Owner: something

[Contact Owner](#)

Description

So delicious, much wow



User Dashboard, a.k.a. “My Stuff”

localhost

WAREHAUS UPLOAD MY STUFF LOGOUT

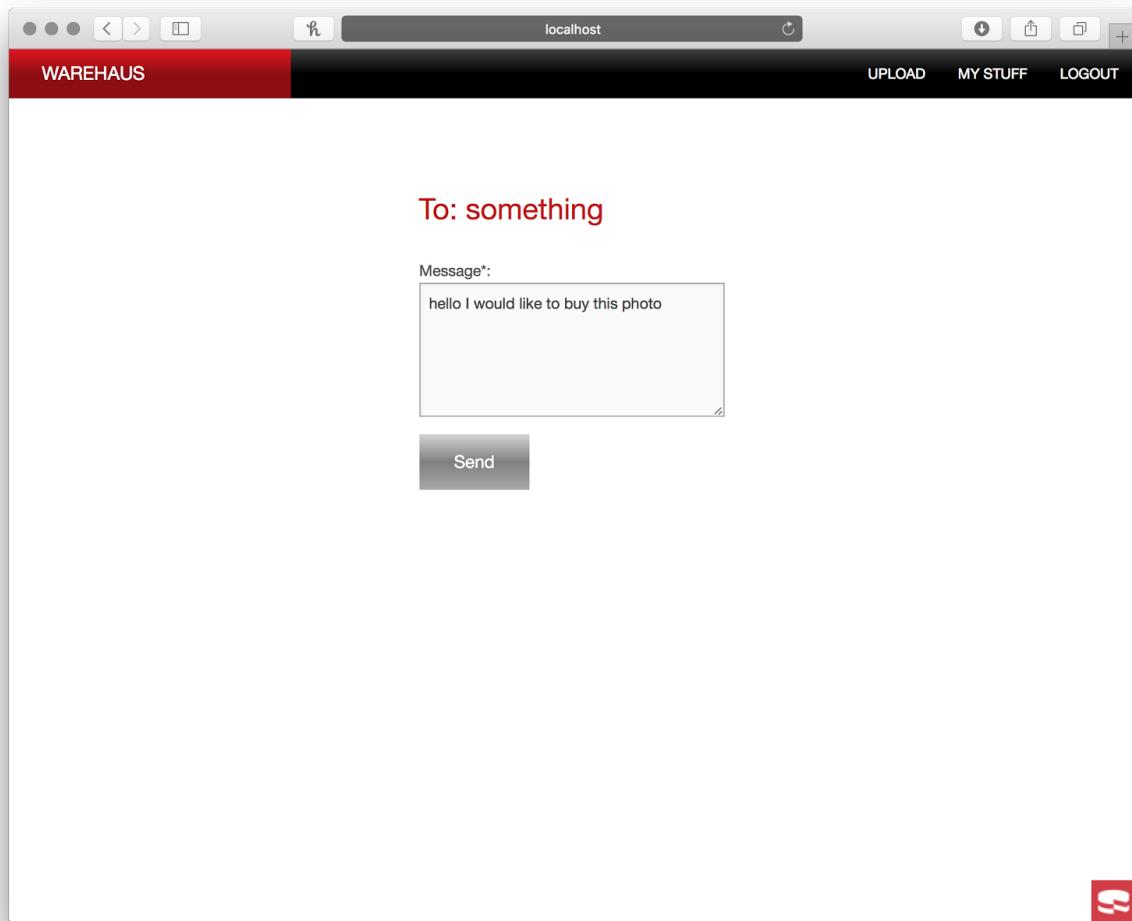
Messages

From:	Message:
yellow	test message
lindsey	hello I would like to buy this photo

My Images

wow.
many edits
so Internet meme
much wiki
very readers
how to article?
such neutral

Message Page



Database Schema

Some developers chose to use MySQL WorkBench, and some chose to work directly within MySQL through the Terminal. The following is the schema dump of the databases, with a sample from the Users table at the end to show that the passwords are encrypted.

Schema Dump

```
-- MySQL dump 10.13 Distrib 5.6.34, for osx10.11 (x86_64)
--
-- Host: localhost      Database: warehaus_media
-----
-- Server version      5.6.34

/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8 */;
/*!40103 SET @OLD_TIME_ZONE=@@TIME_ZONE */;
/*!40103 SET TIME_ZONE='+00:00' */;
/*!40014 SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0 */;
/*!40014 SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS,
FOREIGN_KEY_CHECKS=0 */;
/*!40101 SET @OLD_SQL_MODE=@@SQL_MODE,
SQL_MODE='NO_AUTO_VALUE_ON_ZERO' */;
/*!40111 SET @OLD_SQL_NOTES=@@SQL_NOTES, SQL_NOTES=0 */;

--
-- Table structure for table `Categories`
--

DROP TABLE IF EXISTS `Categories`;
/*!40101 SET @saved_cs_client      = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `Categories` (
  `CategoryID` int(11) NOT NULL AUTO_INCREMENT,
  `Category` varchar(50) DEFAULT NULL,
  PRIMARY KEY (`CategoryID`)
) ENGINE=InnoDB AUTO_INCREMENT=5 DEFAULT CHARSET=utf8;
/*!40101 SET character_set_client = @saved_cs_client */;
```

```

-- 
-- Table structure for table `Media`
-- 

DROP TABLE IF EXISTS `Media`;
/*!40101 SET @saved_cs_client      = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;

CREATE TABLE `Media` (
  `MediaID` int(11) NOT NULL AUTO_INCREMENT,
  `Title` varchar(45) DEFAULT NULL,
  `FileLocation` varchar(45) DEFAULT NULL,
  `ThumbnailLocation` varchar(45) DEFAULT NULL,
  `MediaType` varchar(45) DEFAULT NULL,
  `Format` varchar(45) DEFAULT NULL,
  `DateUploaded` datetime DEFAULT NULL,
  `Price` decimal(11,2) DEFAULT NULL,
  `category_id` int(11) NOT NULL,
  `user_id` int(11) NOT NULL,
  `Description` varchar(255) NOT NULL,
  PRIMARY KEY (`MediaID`,`user_id`),
  KEY `fk_Media_Categories1_idx` (`category_id`),
  KEY `fk_Media_User1_idx` (`user_id`),
  CONSTRAINT `fk_Media_Categories1` FOREIGN KEY (`category_id`)
    REFERENCES `Categories` (`CategoryID`) ON DELETE NO ACTION ON UPDATE
    NO ACTION,
  CONSTRAINT `fk_Media_User1` FOREIGN KEY (`user_id`) REFERENCES
    `Users` (`UserID`) ON DELETE NO ACTION ON UPDATE NO ACTION
) ENGINE=InnoDB AUTO_INCREMENT=10 DEFAULT CHARSET=utf8;
/*!40101 SET character_set_client = @saved_cs_client */;

-- 
-- Table structure for table `Media_Attributes`
-- 

DROP TABLE IF EXISTS `Media_Attributes`;
/*!40101 SET @saved_cs_client      = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;

CREATE TABLE `Media_Attributes` (
  `MediaID` int(11) NOT NULL,
  `Media_Attributes` varchar(1000) DEFAULT NULL,
  PRIMARY KEY (`MediaID`),
  CONSTRAINT `fk_table1_Media2` FOREIGN KEY (`MediaID`) REFERENCES
    `Media` (`MediaID`) ON DELETE NO ACTION ON UPDATE NO ACTION
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

```

```

/*!40101 SET character_set_client = @saved_cs_client */;

-- 
-- Table structure for table `Messages`
-- 

DROP TABLE IF EXISTS `Messages`;
/*!40101 SET @saved_cs_client      = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `Messages` (
  `User1` int(11) NOT NULL,
  `User2` int(11) NOT NULL,
  `Message` varchar(1000) DEFAULT NULL,
  `Date` datetime DEFAULT NULL,
  `MessageID` int(11) NOT NULL AUTO_INCREMENT,
  PRIMARY KEY (`MessageID`),
  KEY `fk_Messages_User1_idx` (`User1`),
  KEY `fk_Messages_User2_idx` (`User2`),
  CONSTRAINT `fk_Messages_User1` FOREIGN KEY (`User1`) REFERENCES `Users` (`UserID`) ON DELETE NO ACTION ON UPDATE NO ACTION,
  CONSTRAINT `fk_Messages_User2` FOREIGN KEY (`User2`) REFERENCES `Users` (`UserID`) ON DELETE NO ACTION ON UPDATE NO ACTION
) ENGINE=InnoDB AUTO_INCREMENT=3 DEFAULT CHARSET=utf8;
/*!40101 SET character_set_client = @saved_cs_client */;

-- 
-- Table structure for table `Transactions`
-- 

DROP TABLE IF EXISTS `Transactions`;
/*!40101 SET @saved_cs_client      = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `Transactions` (
  `OrderDate` datetime NOT NULL,
  `SoldBy` int(11) NOT NULL,
  `PurchasedBy` int(11) NOT NULL,
  PRIMARY KEY (`OrderDate`, `SoldBy`, `PurchasedBy`),
  KEY `fk_Transactions_User1_idx` (`SoldBy`),
  KEY `fk_Transactions_User2_idx` (`PurchasedBy`),
  CONSTRAINT `fk_Transactions_User1` FOREIGN KEY (`SoldBy`) REFERENCES `Users` (`UserID`) ON DELETE NO ACTION ON UPDATE NO ACTION,
  CONSTRAINT `fk_Transactions_User2` FOREIGN KEY (`PurchasedBy`)
  REFERENCES `Users` (`UserID`) ON DELETE NO ACTION ON UPDATE NO ACTION
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

```

```

/*!40101 SET character_set_client = @saved_cs_client */;

-- 
-- Table structure for table `Users`
-- 

DROP TABLE IF EXISTS `Users`;
/*!40101 SET @saved_cs_client      = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `Users` (
  `UserID` int(11) NOT NULL AUTO_INCREMENT,
  `Username` varchar(50) DEFAULT NULL,
  `Password` varchar(255) NOT NULL,
  `Role` varchar(45) DEFAULT NULL,
  `Email` varchar(45) DEFAULT NULL,
  PRIMARY KEY (`UserID`)
) ENGINE=InnoDB AUTO_INCREMENT=7 DEFAULT CHARSET=utf8;
/*!40101 SET character_set_client = @saved_cs_client */;

-- 
-- Table structure for table `phinxlog`
-- 

DROP TABLE IF EXISTS `phinxlog`;
/*!40101 SET @saved_cs_client      = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `phinxlog` (
  `version` bigint(20) NOT NULL,
  `migration_name` varchar(100) DEFAULT NULL,
  `start_time` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
  `end_time` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
  `breakpoint` tinyint(1) NOT NULL DEFAULT '0',
  PRIMARY KEY (`version`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
/*!40101 SET character_set_client = @saved_cs_client */;
/*!40103 SET TIME_ZONE=@OLD_TIME_ZONE */;

/*!40101 SET SQL_MODE=@OLD_SQL_MODE */;
/*!40014 SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS */;
/*!40014 SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS */;
/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
/*!40111 SET SQL_NOTES=@OLD_SQL_NOTES */;
```

```
-- Dump completed on 2017-08-09  3:40:04
```

User Table Data

```
LOCK TABLES `Users` WRITE;
/*!40000 ALTER TABLE `Users` DISABLE KEYS */;
INSERT INTO `Users` VALUES
(4,'something','$2y$10$vgdh9i0EM6tnpPT48P4qIeWoDYZDsN4YULTM/KkP2JyjkK7a
ltn0uG','','passwordissomething@something.com'),
(5,'yellow','$2y$10$ZD0Lcpw79KenTaXQ7dx6F.Z207wp0iEpbgMqfM0mCDPlACT.q
vJe','','yellow'),
(6,'lindsey','$2y$10$YOTidHv2DEWaDKsuN2jnK.kB6aqh4ivqxJ5l/0qV5.4FnI/Hi
YCMO','','lindsey@grandrounds.com');
/*!40000 ALTER TABLE `Users` ENABLE KEYS */;
UNLOCK TABLES;
```

Google Analytics

At the time this document was written, the Google Analytics script was in production code but unable to be accessed for screenshots.

Here is the script call (highlighted) in `src/Template/Layout/default.ctp`:

```
<?php
/**
 * CakePHP(tm) : Rapid Development Framework (http://cakephp.org)
 * Copyright (c) Cake Software Foundation, Inc. (http://cakefoundation.org)
 *
 * Licensed under The MIT License
 * For full copyright and license information, please see the LICENSE.txt
 * Redistributions of files must retain the above copyright notice.
 *
 * @copyright Copyright (c) Cake Software Foundation, Inc.
 * (http://cakefoundation.org)
 * @link http://cakephp.org CakePHP(tm) Project
 * @since 0.10.0
 * @license http://www.opensource.org/licenses/mit-license.php MIT
 * License
 */
?>
<!DOCTYPE html>
<html>
    <head>
        <?= $this->Html->charset() ?>
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
        <title><?= $this->fetch('title') ?></title>

        <?= $this->Html->meta('icon') ?>

        <?= $this->Html->css('base.css') ?>
        <?= $this->Html->css('cake.css') ?>
        <?= $this->Html->css('home.css') ?>

        <?= $this->fetch('meta') ?>
        <?= $this->fetch('css') ?>
        <?= $this->fetch('script') ?>
        <? php include_once("analyticstracking.php") ?>
    </head>
    <body>
        <nav class="top-bar expanded" data-topbar role="navigation">
            <ul class="title-area large-3 medium-4 columns">
                <li class="name">
```

```

        <h1><a href="/pages/home">WAREHAUS</a></h1>
    </li>
</ul>

<div class="top-bar-section">
    <ul class="right">
        <li><a href="/media/add">UPLOAD</a></li>
        <?php if($this->request->session()->read('Auth')) { ?>
            <li><a href="/users/dashboard">MY STUFF</a></li>
            <li><a href="/users/logout">LOGOUT</a></li>
        <?php }
        else{ ?>
            <li><a href="/users/add">REGISTER</a></li>
            <li><a href="/users/login">LOGIN</a></li>
        <?php } ?
    </ul>
</div>
</nav>

<div class="container clearfix">
    <?= $this->fetch('content') ?>
</div>

<footer></footer>
</body>
</html>

```

And the script itself is in `src/Template/Element/analyticstracking.php`:

```

<script>
    (function(i,s,o,g,r,a,m){i['GoogleAnalyticsObject']=r;i[r]=i[r]||function(){
        (i[r].q=i[r].q||[]).push(arguments)},i[r].l=1*new
Date();a=s.createElement(o),
m=s.getElementsByTagName(o)[0];a.async=1;a.src=g;m.parentNode.insertBefore(a,m
)

})(window,document,'script','https://www.google-analytics.com/analytics.js','g
a');

ga('create', 'UA-104316843-1', 'auto');
ga('send', 'pageview');

</script>

```

Team Member Contributions

Below are screenshots of the emails we have sent to each other about our contributions.



Lindsey Anne Hogg

Wed 8/9/2017 10:07 AM

Reply all | ↴

To: Josh Caleb Reed; J H <jhuan319@mail.ccsf.edu>; Jason Wong <jawong1212@gmail.com>
Cc: Jasmine Kaur Mann; Bo Li <bojms45@gmail.com>; Rui Feng ☰

Lindsey:

- served as team lead/CEO
- designed and managed development workflow using GitHub and Trello
- assigned work to team members
- managed team communication through Slack, email, Google Hangouts, and face-to-face conversations
- edited and finalized all documents
- assisted in teaching team development, git etiquette, and daily standups
- built final project skeleton
- wrote script to set up databases
- code reviews, testing, and merging to master branch
- home page search
- 32 commits total

...



Jason Wong <jawong1212@gmail.com>

Wed 8/9/2017 1:29 AM

Reply all | ↴

To: Lindsey Anne Hogg
Cc: Jasmine Kaur Mann; Josh Caleb Reed; J H <jhuan319@mail.ccsf.edu>;
Bo Li <bojms45@gmail.com>; Rui Feng ☰

Jason:

- helped team members better understand cakephp conventions and mvc through meet ups/hangouts video chat
- code review
- parts of milestones documents
- github contributions: 17 commits
- make sure everyone has the up to date database
- deploy final milestone to server

...

BL

Bo Li <bojms45@gmail.com>

Wed 8/9/2017 1:39 AM

To:  Lindsey Anne Hogg ☺



Reply all | v

In the project, I focus more on "message" and "search function" parts.
I complete "message" part, and on "search function", I did some of them by simply
creating a function.

I helped Tyler to do log in page a little bit.
And trying to do log out, but couldn't figure it out.

I used Tyler's laptop and his Github account to push all codes that I did.

Bo Li

...

BL

Bo Li <bojms45@gmail.com>

Wed 8/9/2017 1:40 AM



| v

BTW, I clean up "Mystuff" to looks nicely by creating table to display the user who
from, and message that the user sent

...



Josh Caleb Reed

Wed 8/9/2017 6:56 AM



Reply all | ▾

To: J H <jhuan319@mail.ccsf.edu>; Jason Wong <jawong1212@gmail.com>

Cc: Lindsey Anne Hogg; Jasmine Kaur Mann; Bo Li <bojms45@gmail.com>; Rui Feng

Josh:

- database construction
- vertical prototype construction
- media authentication
- thumbnail conversion
- logout function
- all milestone docs

Get [Outlook for iOS](#)

...
...



J H <jhuan319@mail.ccsf.edu>

Wed 8/9/2017 2:50 AM



Reply all | ▾

To: Jason Wong <jawong1212@gmail.com>
Cc: Lindsey Anne Hogg; Jasmine Kaur Mann; Josh Caleb Reed; Bo Li <bojms45@gmail.com>; Rui Feng

Tyler:

- contribute media resources
- parts of mockups
- parts of milestones documents
- navigate bar and parts of template default page
- assisted teammates when needed
- created development pages
- collaborated with Bo when he had troubles setting up

...
...

 JM

Jasmine Kaur Mann

Wed 8/9/2017 2:31 AM



Reply all | ▾

To: Jason Wong <jawong1212@gmail.com>; Lindsey Anne Hogg

Cc: Josh Caleb Reed; J H <jhuan319@mail.ccsf.edu>; Bo Li <bojms45@gmail.com>; Rui Feng

Jasmine:

- worked on the navbar/template default page
- worked on mockups and general view of website
- parts of the milestone docs
- github contributions: 6 commits

...
 RF

Rui Feng

Wed 8/9/2017 2:39 AM



Reply all | ▾

To: Jason Wong <jawong1212@gmail.com>
Cc: Lindsey Anne Hogg; Jasmine Kaur Mann; Josh Caleb Reed; J H <jhuan319@mail.ccsf.edu>; Bo Li <bojms45@gmail.com> **Raymond:**

- developed page that display detail of a posted image and assign url to the page
- developed first working prototype and final modification of the user dashboard
- helped Tyler and Bo use git to push and publish new branches onto Github repo
- helped Bo understand how to configure url generations in order to assign controller to pages
- helped team to learn how cakePHP framework automatically generate mySQL queries and its MVC structure and usage
- tested and made suggestions on modifying our database so that it followed cakePHP conventions to speed up developing process by making better usage of the framework functionalities
- helped team learn how to pass information between pages through url generations
- Github commits: 9 commits

...


Project Challenges and Resolutions

This project was a great exercise in learning how to truly work on a software engineering team. The main challenge we had, as with any project, was communication. Many of us have multiple scheduling conflicts outside of class times which prevented us from finding a recurring time to meet as a team. We overcame this by scheduling work meetings on the fly at different times, meeting in class when we could, and primarily using Slack as a form of communication.

Most of the challenges we faced have already been outlined in the “Key Risks” section of Milestone 2. We overcame the skills risk with knowledge transfer between team members who had more experience in various areas helping team members who had less experience. The scheduling risk was mitigated by using asynchronous development and increasing communication when needed. We also reduced the scope of the final project by a lot to account for the time crunch towards the end.

Going back, we could have improved our communication even more by setting a scheduled standup earlier in the project. We also could have chosen a different framework that is better documented and easier to work with.

In the end, though, the team was able to come together, help each other out, and complete the project on time. That’s the most important takeaway--learning how to work as a team.