Final Project for SW Engineering CSC648/848 Section 01 Summer 2017

Local Team

Team Number 04

“PixSale”

Manuel Duran   jdceren@mail.sfsu.edu
Stephen Josey SFSU Student
Ryan Jung SFSU Student
Youssef Hakkou SFSU Student
Jeremy Tan SFSU Student
Yoezhou Yu SFSU Student

Milestone 5

URL: http://sfsuse.com/~su17g04/

08/09/2017

Product summary

PixSale

1. Guests can browse and search through media without being registered users.
2. Users can browse through media by category.
3. Users are able to register and login to their own accounts.
4. Passwords are encrypted on the database.
5. All user information is stored on the database.
6. Registered users are able to upload photos and videos to the website.
7. Registered users are able to make purchases on the website.
8. Admins are able to remove any media from the database.
9. Admins are able to remove any user from the database.

URL of the system to be tested: http://sfsuse.com/~su17g04/

SW Engineering CSC648/848 Section 01 Summer 2017

“PixSale”

Team Number 04

Manuel Duran jdceren@mail.sfsu.edu
Stephen Josey
Ryan Jung
Youssef Hakkou
Jeremy Tan
Yoezhou Yu

Milestone 1

06/30/2017

Revision History

Version	Action	Approval	Date
1.0	Creation of original Milestone 1 document for submission	Team Lead/Editor	06/30/2017
2.0	Updated Milestone 1 with professor’s feedback	Team Lead/Editor	07/03/2017

Content and structure for Milestone 1 document for review

1- Executive Summary

In today's modern world, product advertising requires heavy use of media technology to capture good, quality images that are used to sell a product. This necessity to continuously provide new and unique quality images, that can relate to a market audience, could potentially become a lucrative form of side work or hobby for people who already invest a good amount of their time capturing images. As people spend more and more of their time online, many could benefit from selling videos and pictures they take on their daily basis to buyers or agencies. To this end, pioneers that provides such platforms and services to sell such images could profit from such sales. Our project consist of creating a website where people can upload their images for sale and for potential customers to browse and buy the images they may want or need. Some competitors such as Etsy.com provides this service at a very attractive price. They keep 3.5% of the sale price, which is one of the lowest and least complicated sale's structure out there. However, because they sell many more products than just pictures, this section is not easy to browse and find on their site and people may not even know Etsy sells photos.

Our website would be entirely dedicated to sell videos and pictures which would make it much easier to browse for them. We will emulate Amazon's search engine to provide customers with a quick and easy way to find desired photos/videos, thus providing a great and satisfying customer experience. Our price commision initially would probably be similarly low to Etsy in order to gain traffic and make it attractive for people to sell their images.

We are students from San Francisco State University and the tentative name for our project is PixSale. We are small team of students operating as a small startup while learning and applying basic software engineering principles.

2- Use Cases

Guest - John navigates to PixSale's homepage to browse some **media**. He immediately notices how easy it is to browse and search through the categories to find his favorite pictures of the beach. He chooses an image and is shown a larger sample of the image, with a short description and price on the side. John adds it to his **cart** with one simple button, and after browsing a short while longer decides to choose check out. Upon doing so, he is prompted with an option to become a registered user or to continue browsing. He decides to become a **registered user** so he can continue buying pictures later on for his portfolio, and is prompted for his full name and email, and it was as easy as that: John is registered.

Registered User - Laura goes to PixSale's homepage looking to sell some of her latest videos. Laura is already registered, so she logs in so she has access to her account. She then uploads her videos and sets a price and description for each video. The videos are immediately added to PixSale's entire catalog and can be found by other users. Laura also decides to buy an image of her favorite car so she quickly searches and finds a really nice image and places it in her cart. If Laura was not registered, she is only prompted to register after trying to checkout items from her cart. Since she's already registered, she decides to check out and is able to purchase the image immediately.

Administrator - Steve administers the PixSale website. He notices some inappropriate **media** has been uploaded on the site. Steve launches mySQL Workbench and logs into the PixSale database quickly. He does a quick select in SQL and finds the inappropriate ID of the item that has been uploaded. Steve proceeds to run a delete command in order to remove it from the database. Steve can also remove the user if the user has posted inappropriate content more than once.

3- Data Definition

Guest – Only able to search and browse images. Must register before uploading or purchasing media.

Registered User – Able to search, browse, upload, and purchase media. Must login before uploading and purchasing media. Must provide a valid name and email to create an account.

Admin – Able to access and modify the database. Must login before accessing database. Admin also has the same rights as a registered user.

Media – Media falls under

1) Images – Must contain title, thumbnail, category, price, registered user's name, brief description, and dimensions (length and width in pixels)

2) Videos – Must contain title, thumbnail, category, price, registered user's name, brief description, and video length (must only use minutes and seconds)

Account – A registered user's account. Account must have a username and password as well as a valid name and email.

Cart - A list of potential media that a user would like to purchase.

Database – Stores account information of registered users and admins. Stores images and videos that registered users upload.

4- Initial list of functional requirements

1. Guests shall be able to browse through media by category.
2. Guests shall be able to view descriptions and titles of media such as images and videos.
3. Guests shall be able to search through media without registering.
4. Guests shall be required to provide a valid name and email address upon registration.
5. Registered users shall be able to upload images and videos to the database.
6. Registered user’s account information such as username, password, name, and email shall be stored on the database.
7. Registered users’ passwords shall be encrypted on the database.
8. Guests shall be able to add media items to a cart as well as remove media items from the cart.
9. Registered users shall be able to purchase media.
10. Admins shall be required to login before accessing the database.
11. Admins shall be able to remove any media from the database.
12. Admins shall be able to remove any user from the database

5- List of non-functional requirements

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, Spring 2017. For Demonstration Only". (Important so as to not confuse this with a real application).

6- Competitive analysis

Key Features	PixSale	iStock	Dreamstime	GettyImages
Dropdown list for related search and suggestions	A list that tries to predict what the user is searching for	Shows a short list of predicted words	Shows predicted words and also suggestions for categories	Shows a short list of predicted words
Sorting photos and pictures into categories	Give each photo tags and keywords in order to sort them	Photos are sorted by tags and keywords	Photos are sorted by tags and keywords. Also split by category	Photos are sorted by tags and keywords
Registering	Registering by entering an email and password	Simple and easy way to register by using email	All you need is a email and password. Very simple	Able to sign up using facebook with one click
Easy payment method	Add photos into a cart and pay upon checkout with credit card	Buy each photo individually after clicking the “buy” button	Buy each photo individually after clicking the “buy” button	Add photos into a cart and pay upon checkout with credit card

One of the advantages of our product compared to what’s already out there on competitive products is that our search function will try to emulate Amazon’s search function where it will show a list of predicted words and also the category that we are trying to search for. Next, the way we plan to sort our photos is by giving each photo multiple keywords and tags to make them easily findable, then we are going to place each photo in a category that consists of the same types of photos. Registering will be as simple as possible. One way is to incorporate sign-ups with Google or Facebook to make it a one-click process as compared to sites like iStock and Dreamstime that require entering our emails. For the payment method, we are going to try and go down the GettyImages route by having a shopping cart that holds all photos that the user wants to purchase and have them check out everything in one go.

7- High-level system architecture.

Framework

- CakePHP: We are choosing to use CakePHP because it is open-source and easy to use. There are also a lot of resources to help us if we get stuck. CakePHP is also updated regularly so it should have a long lifespan. It uses the MVC architecture and is fast and flexible.

GUI technologies

- Bootstrap: This GUI technology helps to provide the user with an incredible user experience no matter what device he or she uses.
- Javascript: It is well supported with almost every internet browser. It is easy to learn and will prove useful.
- CSS: We will use css to style our web pages and make it visually pleasant for the user.
- Jquery: We will use this javascript library to simplify javascript code.

Supported Browsers

Chrome v59.0.3071 to 61.0,
Firefox v54.0-66.0a1
Safari v10.1

API's

Stormpath API: This API supports Facebook and google login for website registration.

8- Team

Manuel Duran - Chief Executive Officer
Stephen Josey - Chief Technology Officer
Ryan Jung - Backend developer
Youssef Hakkou - Backend developer
Jeremy Tan - Frontend developer
Yuezhou Yu - Frontend developer

9- Checklist

Team decided on basic means of communications
DONE. RECOMMEND TEAM GETTING THE SLACK APP
Team found a time slot to meet outside of the class
DONE
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

SW Engineering CSC648/848 Section 01 Summer 2017

"PixSale"

Local Team
Team Number 04

Manuel Duran jdceren@mail.sfsu.edu
Stephen Josey SFSU Student
Ryan Jung SFSU Student
Youssef Hakkou SFSU Student
Jeremy Tan SFSU Student
Yoezhou Yu SFSU Student

Milestone 2

07/14/2017

Revision History

Version	Action	Approval	Date
1.0	Creation of original Milestone 2 document for submission	Team Lead/Editor	7/14/2017
2.0	Updated Milestone 2 with Professor's feedback	Team Lead/Editor	7/19/2017

Content and structure for Milestone 2 document for review by institutors

1. Use Cases V2

Guest - John navigates to PixSale's homepage to browse some media. He immediately notices how easy it is to browse and search through the categories to find his favorite pictures of the beach. He chooses an image and is shown a larger sample of the image, with a short description and price on the side. John adds it to his cart with one simple button, and after browsing a short while longer decides to choose check out. Upon doing so, he is prompted with an option to become a registered user or to continue browsing. He decides to become a registered user so he can continue buying pictures later on for his portfolio, and is prompted for his full name and email, and it was as easy as that: John is registered.

Registered User - Laura goes to PixSale's homepage looking to sell some of her latest videos. Laura is already registered, so she logs in so she has access to her account. She then uploads her videos and sets a price and description for each video. The videos are immediately added to PixSale's entire catalog and can be found by other users. Laura also decides to buy an image of her favorite car so she quickly searches and finds a really nice image and places it in her cart. If Laura was not registered, she is only prompted to register after trying to checkout items from her cart. Since she's already registered, she decides to check out and is able to purchase the image immediately.

Administrator - Steve administers the PixSale website. He notices some inappropriate media has been uploaded on the site. Steve launches mySQL Workbench and logs into the PixSale database quickly. He does a quick select in SQL and finds the inappropriate ID of the item that has been uploaded. Steve proceeds to run a delete command in order to remove it from the database. Steve can also remove the user if the user has posted inappropriate content more than once.

2. Data Definitions V2

Guest – Only able to search and browse images. Must register before uploading or purchasing media.

Registered User – Able to search, browse, upload, and purchase media. Must login before uploading and purchasing media. Must provide a valid name and email to create an account.

Admin – Able to access and modify the database. Must login before accessing database. Admin also has the same rights as a registered user.

Media – Media falls under

1) Images – Must contain title, thumbnail, category, price, registered user's name, brief description, and dimensions (length and width in pixels)

2) Videos – Must contain title, thumbnail, category, price, registered user's name, brief description, and video length (must only use minutes and seconds)

Account – A registered user's account. Account must have a username and password as well as a valid name and email.

Cart - A list of potential media that a user would like to purchase.

Database – Stores account information of registered users and admins. Stores images and videos that registered users upload.

3. Functional Requirements V2

Priority 1 requirements:

1) Guests shall be able to view descriptions and titles of media such as images and videos.
2) Guests shall be able to search through media without registering.
3) Guests shall be required to provide a valid name and email address upon registration.
4) Guests shall be able to browse through media by category.
5) Registered users' passwords shall be encrypted on the database.
6) Registered user's account information such as username, password, name, and email shall be stored on the database.
7) Registered users shall be able to upload images and videos to the database.
8) Registered users shall be able to purchase media.
9) Registered users shall have the same rights as a guest user
10) Admins shall be required to login before accessing the database.
11) Admins shall be able to remove any media from the database.
12) Admins shall be able to remove any user from the database.
13) Admins shall have the same rights as a registered user.

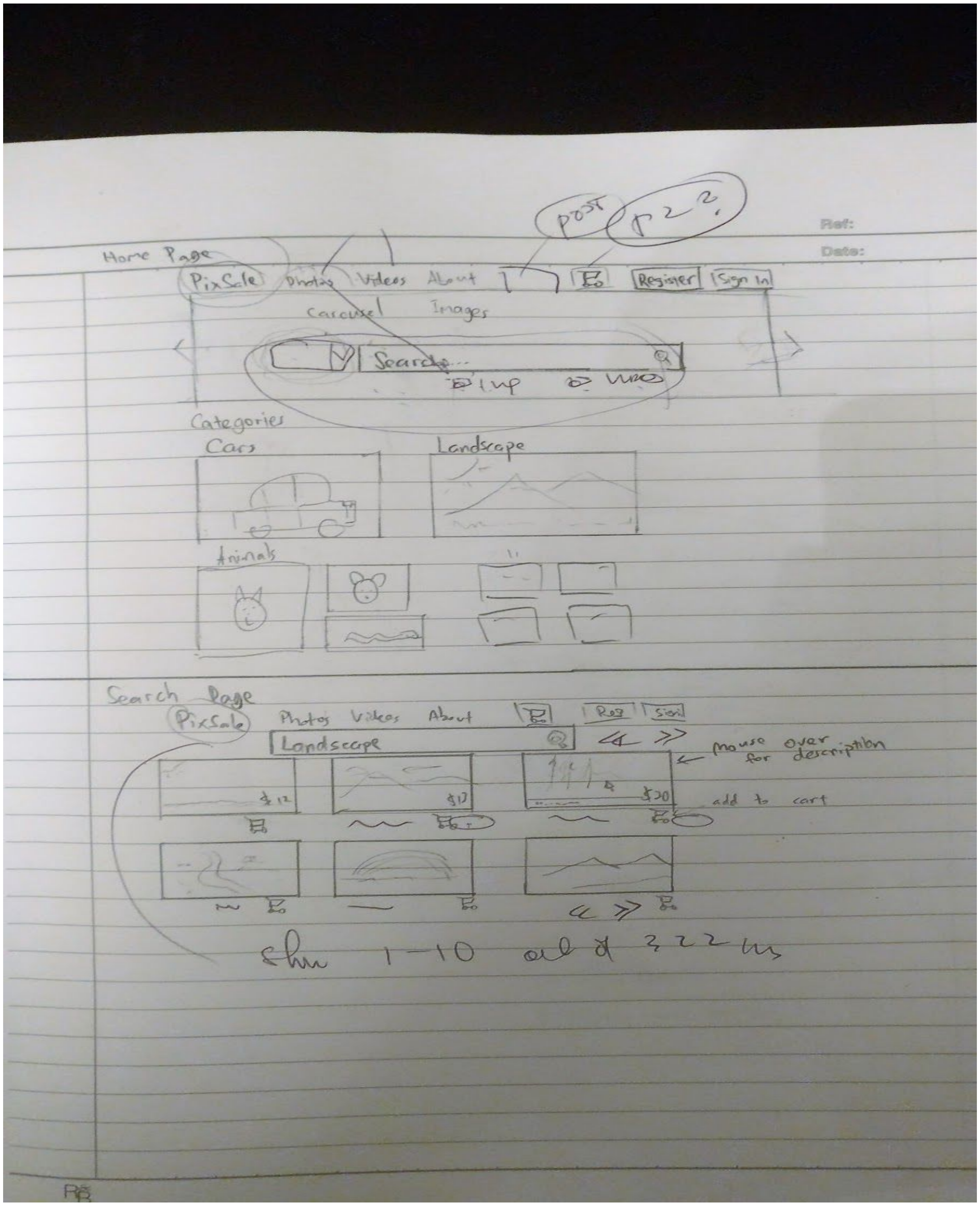
Priority 2 requirements:

14) Guests shall be able to add media items to a cart as well as remove media items from the cart.

4. Non-Functional Requirements V2

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, Spring 2017. For Demonstration Only". (Important so as to not confuse this with a real application).

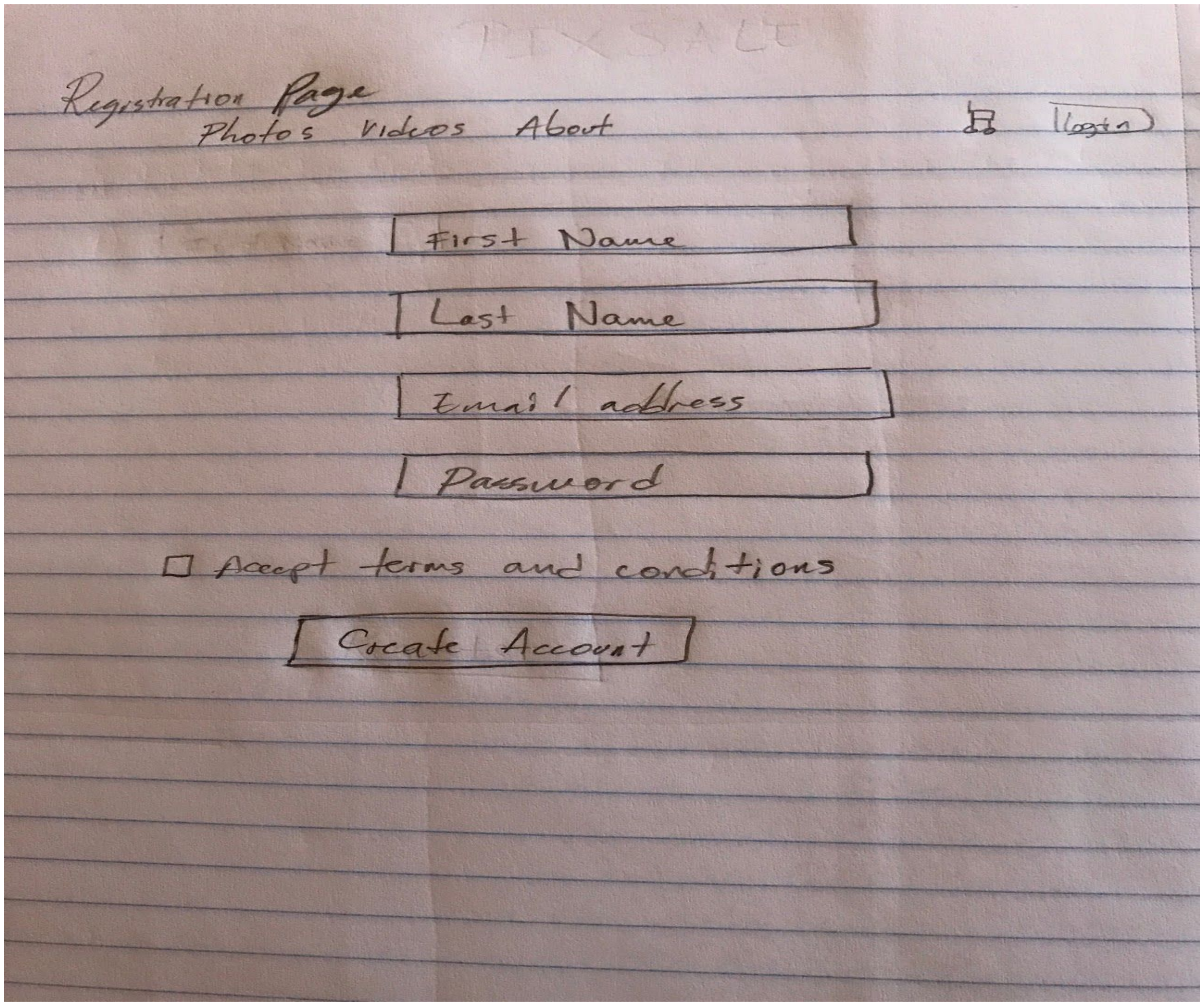
5. UI Mockups and Storyboards (high level only)



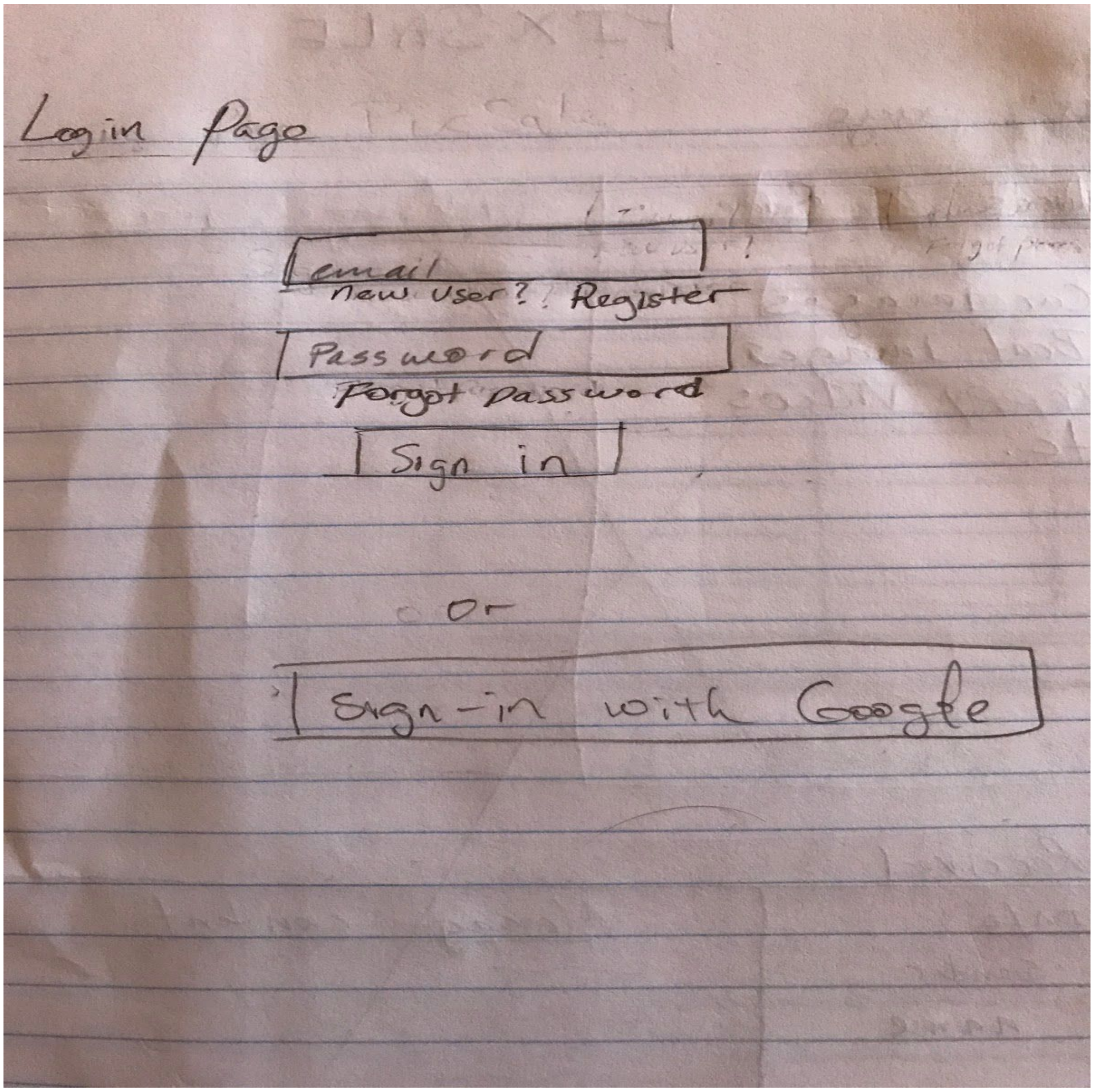
The **guest** will begin at the Home Page where he is able to view previews of **media** from different categories. The guest will be able to filter the search for images or videos by using checking checkboxes near the search bar. The guest can also choose any of the available categories in a drop down menu next to the search bar.

Once the **guest** enters a keyword to search, the user will be redirected to a search page where they can view any **media** whose category, media type, and title matches the user's input. On the search page the guest will see the thumbnails and titles of media and will be able to click on any media that interest them to purchase it.

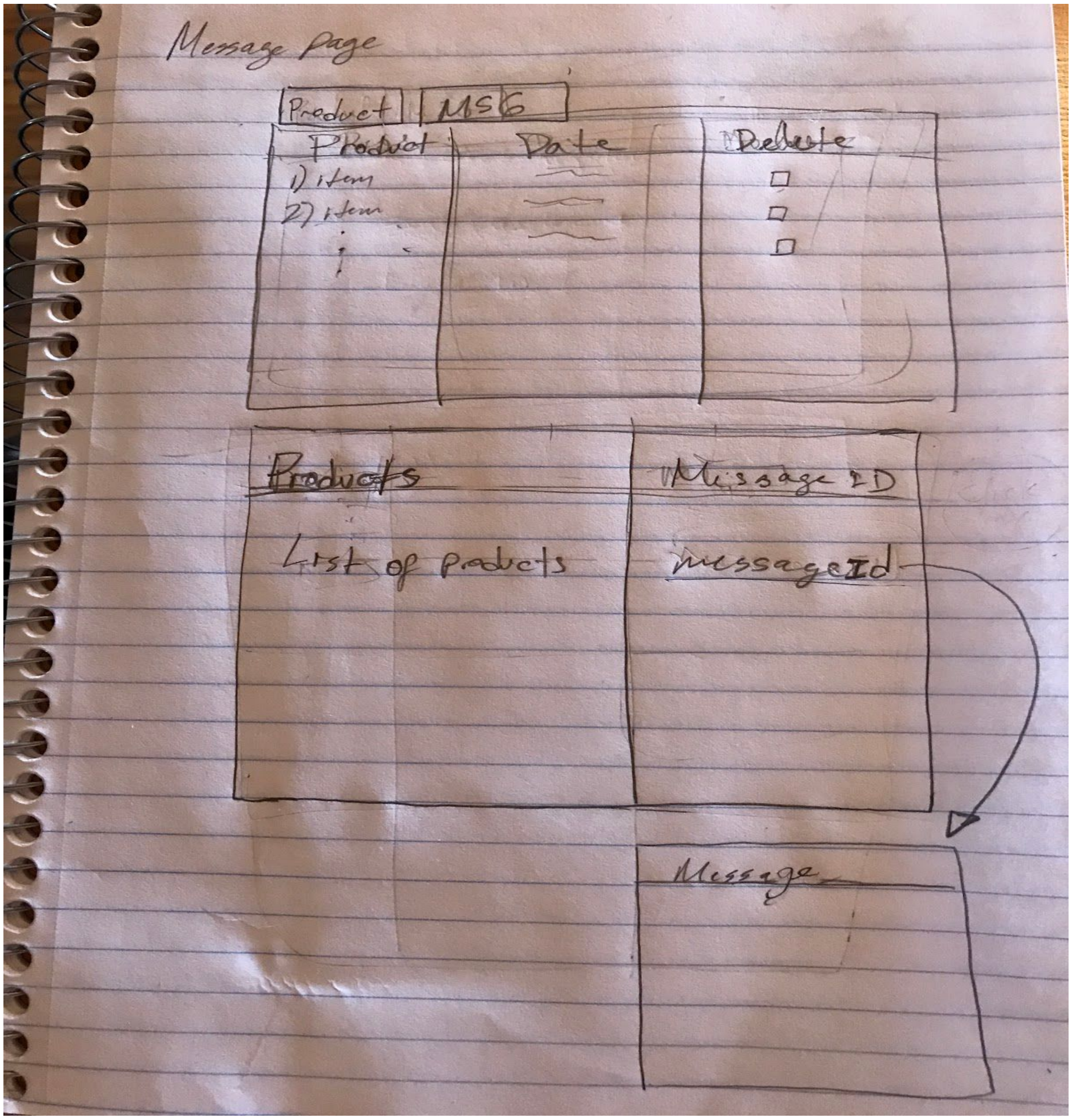
The search page will also display how many media items are currently being displayed as well as how many search results matched the **guest's** query. The **guest** will only be display small amounts of media items at a time and will be able to browse through more media items using navigation arrows.



This is the registration page. All users who wish to buy or sell will have to become a **registered user**. This page can be accessed at any time via a button in the top-right of the web page, on any page. It will also be shown to any **guest** who is attempting to complete a purchase. The page will require a first name, last name, email address, username, and a password. The password will be encrypted before being inserted into our database. We will also have the option of signing in via Google to save users some time. Once a user registers, they will be able to login to the site. Any user that is **registered** and logged in will be able to complete purchases (which will send a message to the seller), or sell their own **media** items.



The login page will appear when the **guest** clicks the login button on the navigation bar or when a guest wants to purchase **media**. If the user does not have an account on PixSale or a Google account they will be asked to register before continuing with their purchase. Once the **guest** has logged in he will become a **registered** user and will be able to continue on in the purchasing processes. After confirming the purchase a message will be sent to the appropriate **registered user** who uploaded the media. The uploader will then see a message about a purchase request in his mailbox and the uploader can work with the buyer to complete the purchase.

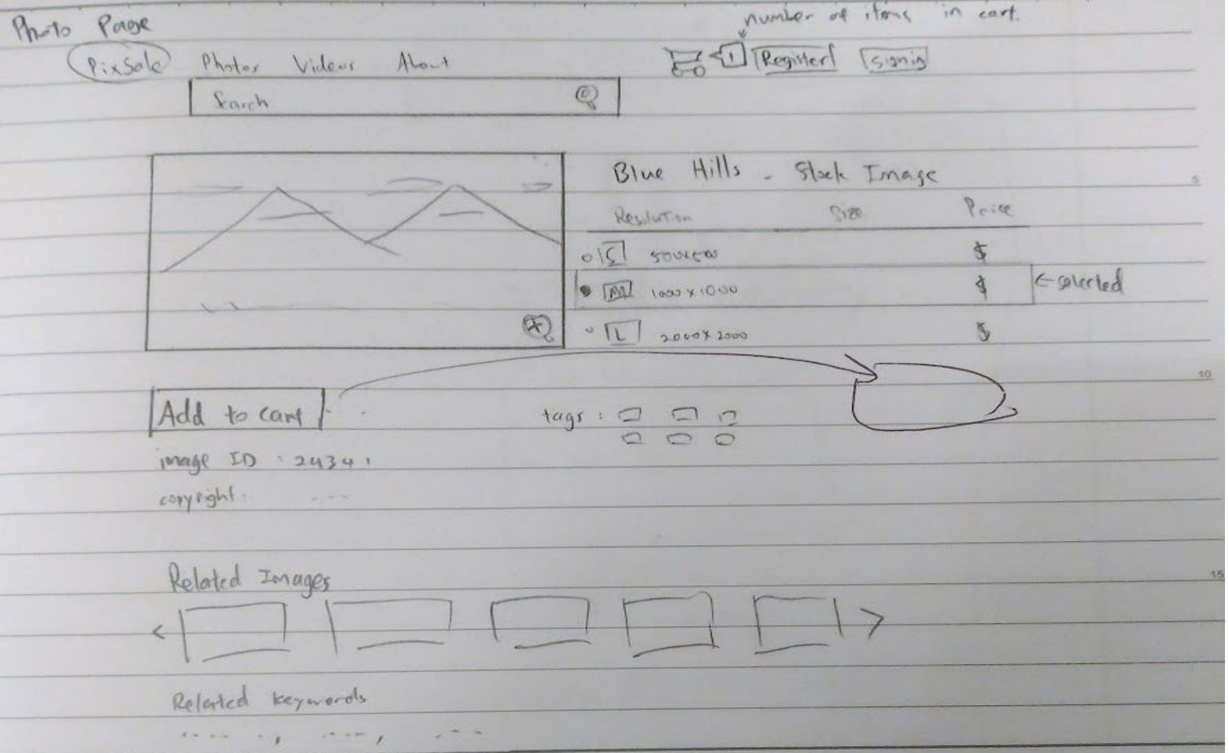


This is the **message** page. When a **registered user** requests a purchase of a **media item**, it will always send a **message** to the seller to inform them of the purchase. The seller may log in at any time and view his/her own **messages** to see who has bought the items they have put up for sale. All **messages** will be linked to a sender as well as an item in which the **message** is concerning. Sellers can delete **messages** after reading them, if they desire.

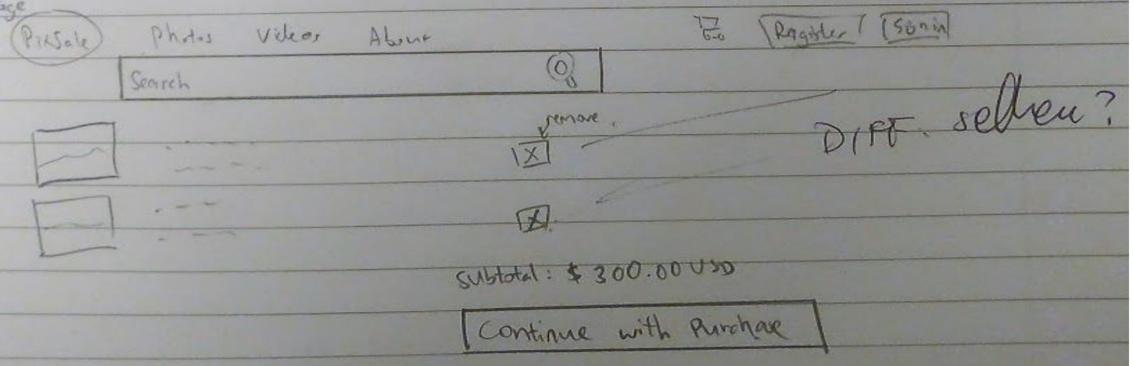
Ref:

Date:

Photo Page



Cart Page



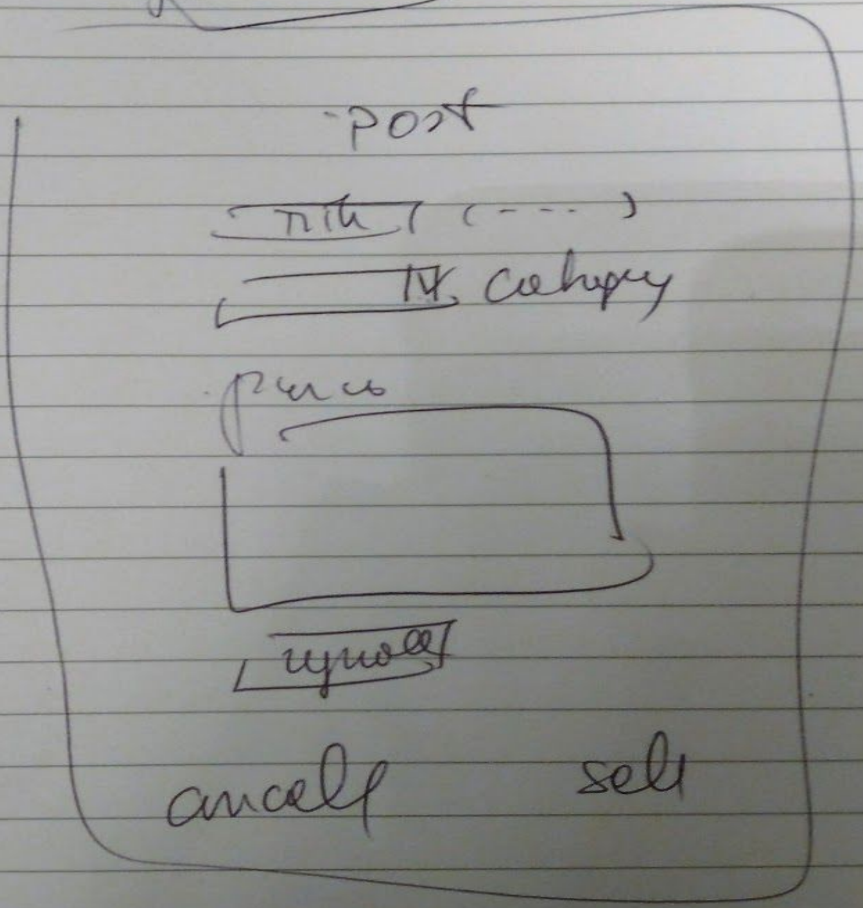
While browsing, any **media** item can be clicked in order to show more details about it. This will typically be the second stage taken by a buyer. However, any user will be able to drill down into a **media** item by clicking on it (the top image), which will show further details including a larger sample, different qualities, and the ability to be able to add it to a cart. This detailed view will also show related items as suggestions. All samples will be low quality so the content simply can't be taken from the site. From this page the user can choose to go back to their search in order to keep browsing, or view their cart.

After adding a **media** item to their cart, the user can view their cart (the bottom image) in order to remove items if they choose, or proceed to purchase everything in their cart. For our purposes, the purchase will simply send the **seller** a **message** informing them that the buyer wants to buy their item(s). If the buyer is not a **registered user**, they will be forced to register before finalizing the purchase.

Ref:

Date:

Post sell



This is the “Sell” or upload page. Only **registered users** may access this page, after logging in with their username and password. Using this page, **registered users** will be able to post new **media** items. The new **media** items will immediately be posted for other users to browse. They must include a title, category, price, and description of the item before being able to post it.

This page is connected directly to our **database.** All the fields are corresponding fields for the tables in our database, and will include actually uploading the **media** item to the server in a certain directory we’ve chosen.

All **media** items posted by a user will be visible to them in their own account, so they can choose to remove them at any time.

6. High level Architecture, Database Organization

Frameworks:

Cakephp
Bootstrap
Jquery

MySql Database Tables:

Registered Users - user id, username, password, email, first name , and last name

Messages - sender's user id, receiver's user id, message title, message

Categories - category id, category name

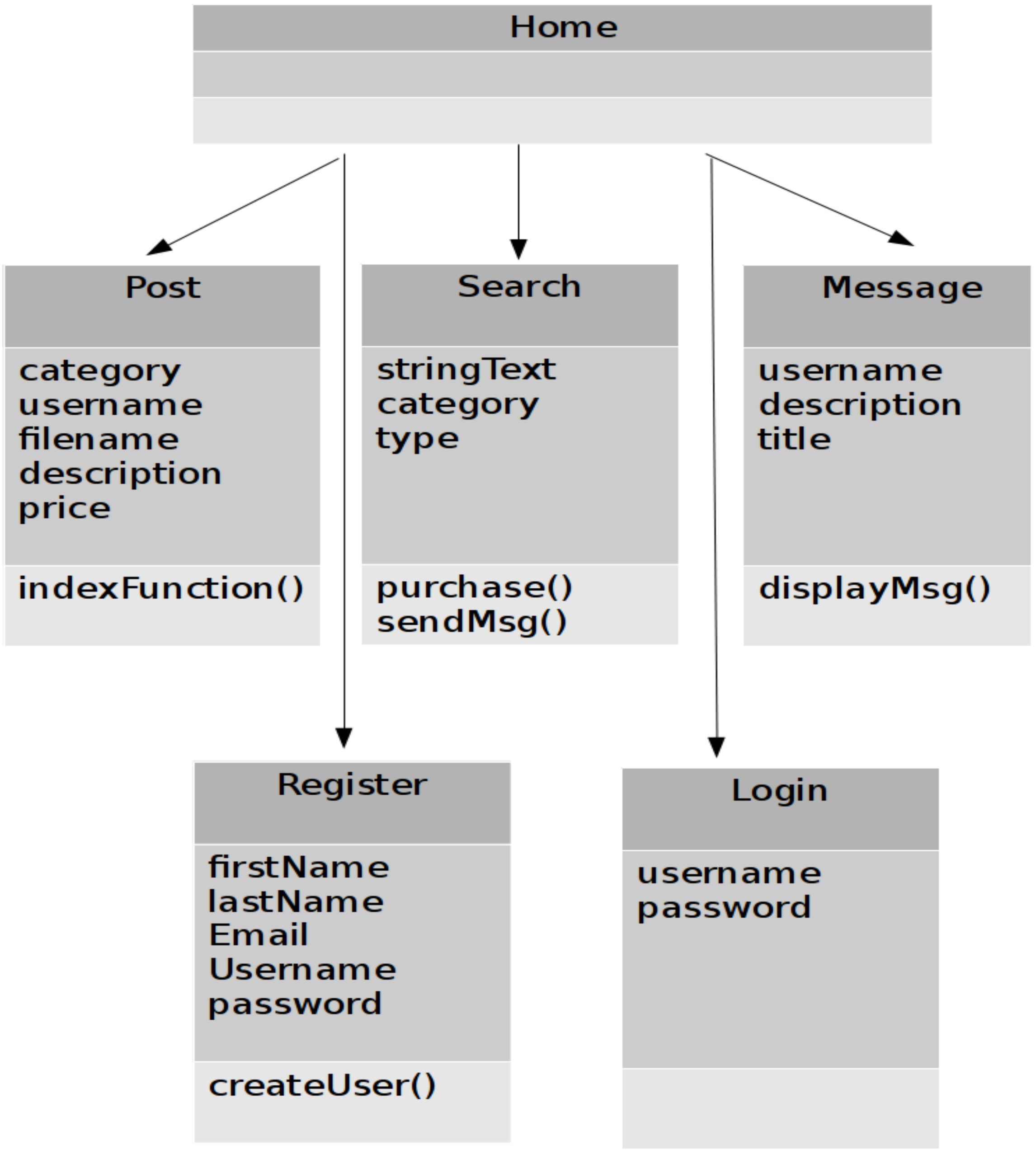
Media items - title , description , price, category, user id , media type, file path, video length, image size

We will be using a file system to contain images and videos. We will be supporting frequently used image formats such as JPG and PNG and the MP4 format for videos.

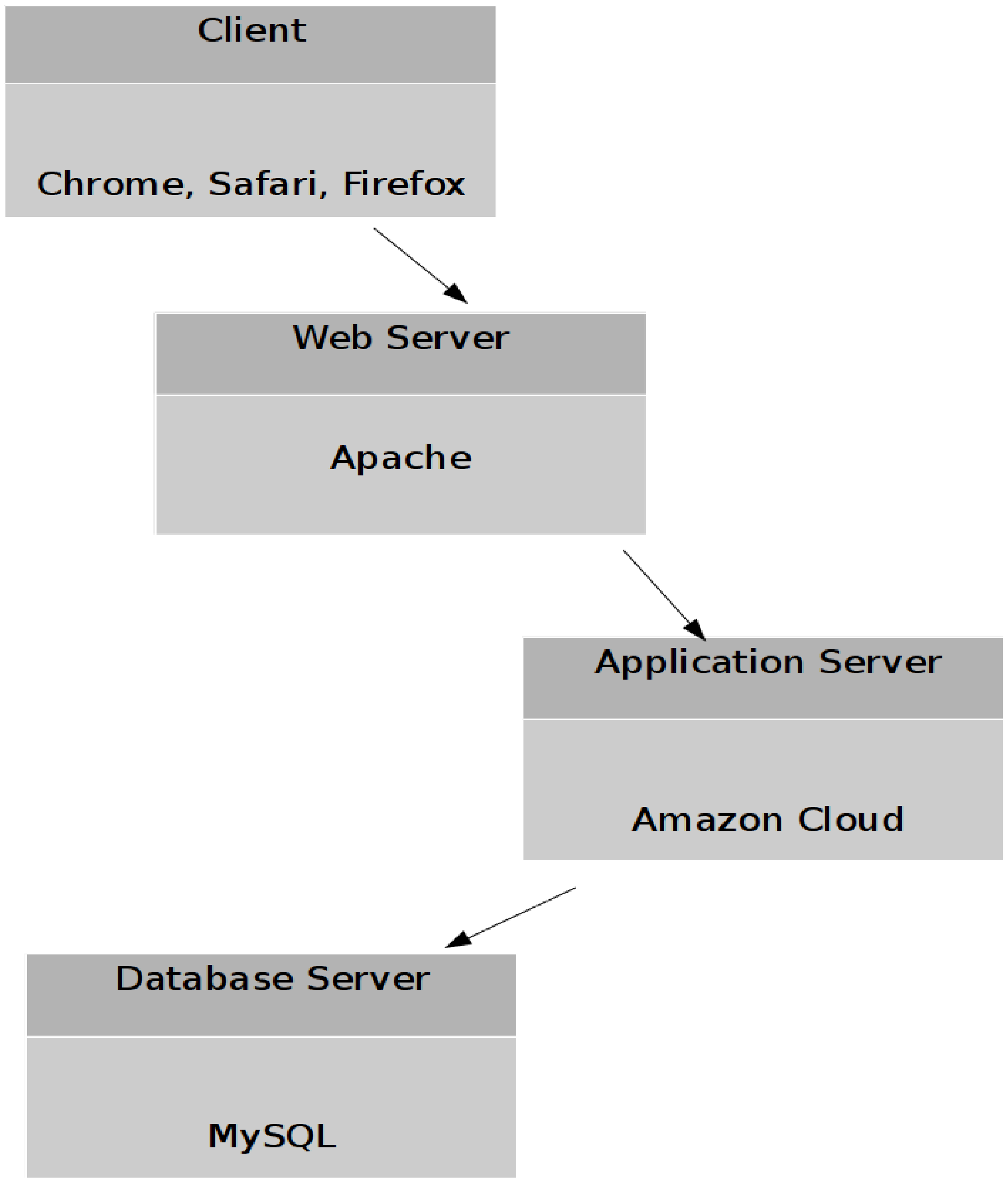
To search for media items we will search against the category, title, and media type in *Media items*. The user will select a category and media type and can choose to enter a key term in the search bar. The search will then query *Media Items* table to find any records whose title contains the user's key term and whose category and media type match the user's input. We will use "LIKE %".$search."%" search type instead of SQL exact search in order to avoid writing the entire title to get a return result. It will also search on on all fields as well.

7. High Level UML Diagrams

High-level UML class diagrams for implementation classes of core functionality.



UML Component and deployment diagrams



8. Key Risks

Skills Risks
The main risks we have here is that none of our members have ever used CakePHP so we've all been having to learn it as we go along via tutorials. Very few of us have database or PHP experience, so needing to learn the CakePHP framework as well as using MySQL to handle our database needs has slowed down our workflow. We've been able to overcome these risks by using online tutorials from Lynda.com as well as simply helping each other understand certain things.

Schedule Risks
We are able to schedule meeting when needed without much trouble. We tend to meet before class and communicate via Slack. The only risk comes in if we need to do an in-person meeting that is not before class because some of us live in San Francisco, and some of us live in the East Bay, so it's hard find times to meet up that is convenient for everyone. We will overcome this by agreeing to do a meeting in San Francisco one time, and the East Bay another time.

Technical Risks
No technical risks have arisen so far. We've been able to overcome installing the main CakePHP framework on the server-side, as well as setting up the database. We've also managed to use GitHub decently well, using different branches for different features/milestones. There is one branch dedicated to "deployment" so no one overwrites it except the CTO when merging other changes.

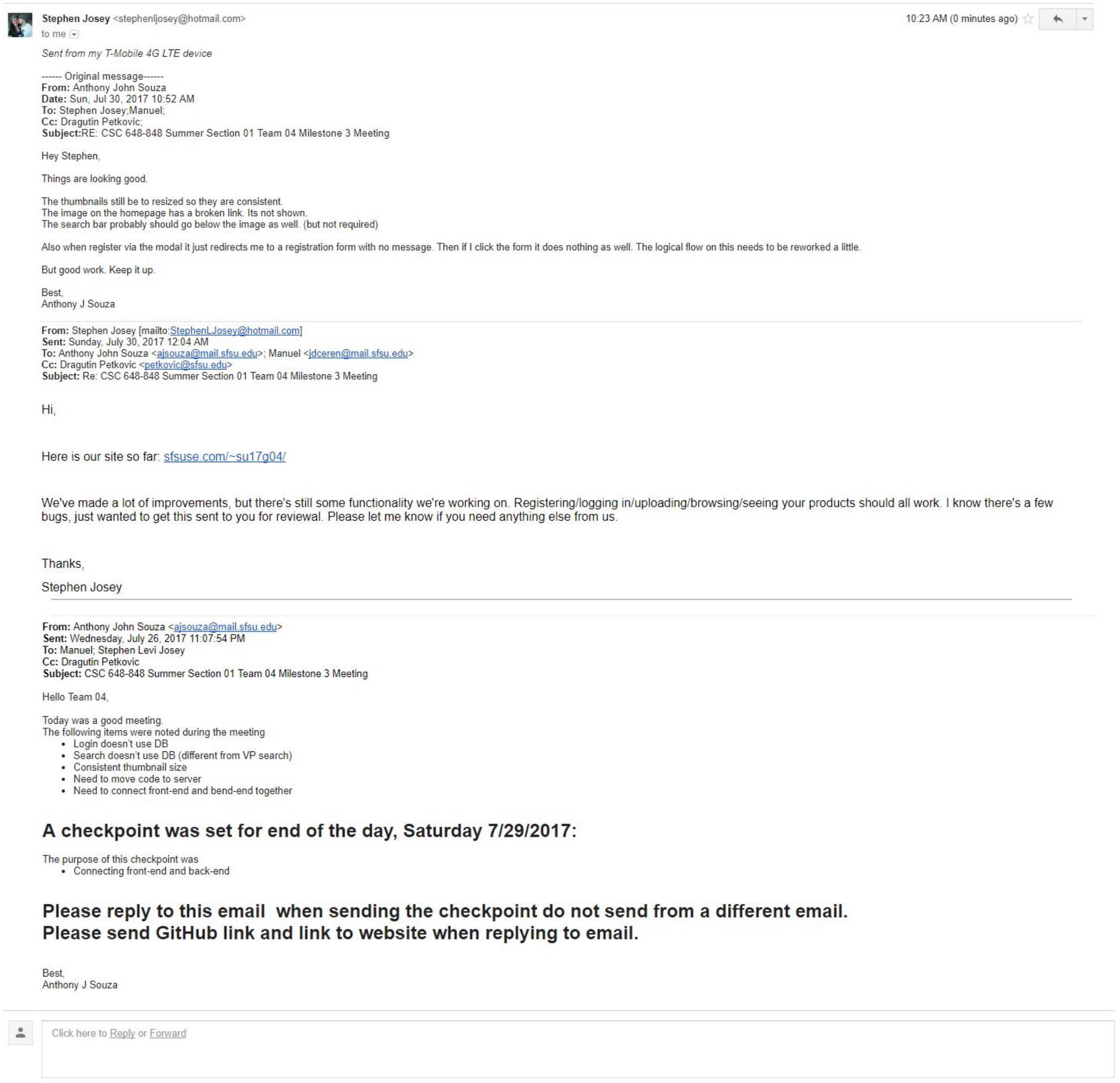
Teamwork Risks
There has been no teamwork risks that have arisen so far. We get along well with each other, and so far have voiced our opinions and hear one another out.

Legal/Content Risks
None. All content will be taken by our photographer, so we do not have to worry about copyright issues.

9. Team Organization

Manuel Duran - Our team leader and CEO. Manuel’s main role is to organize different tasks and meetings, and to make sure we all keep up. Manuel weighs in on any non-technical decisions that need to be made.
Stephen Josey - Our CTO. Stephen’s main role is to make sure all technical questions are answered. All installations and setup are overseen by Stephen.
Ryan Jung - Our backend developer. Ryan has experience with working with the database and connecting PHP code to it.
Youssef Hakkou - One of our frontend developers. Youssef lays out all the HTML/CSS code on the page based on designs we’ve created.
Jeremy Tan - Our UI designer. Jeremy plays a key role in the layout of the site, and helping the frontend developers code to the design he has made.
Yoezhou Yu - Another one of our frontend developers.Youezhou helps lay out all the HTML/CSS code on the page based on designs we’ve created.

Milestone 3 Feedback Summary



Stephen Josey <stephenljosey@hotmail.com>
to me

10:23 AM (0 minutes ago)

Sent from my T-Mobile 4G LTE device

------ Original message------
From: Anthony John Souza
Date: Sun, Jul 30, 2017 10:52 AM
To: Stephen Josey;Manuel;
Cc: Dragutin Petkovic;
Subject:RE: CSC 648-848 Summer Section 01 Team 04 Milestone 3 Meeting

Hey Stephen,

Things are looking good.

The thumbnails still be to resized so they are consistent.
The image on the homepage has a broken link. Its not shown.
The search bar probably should go below the image as well. (but not required)

Also when register via the modal it just redirects me to a registration form with no message. Then if I click the form it does nothing as well. The logical flow on this needs to be reworked a little.

But good work. Keep it up.

Best,
Anthony J Souza

From: Stephen Josey [mailto:StephenLJosey@hotmail.com]
Sent: Sunday, July 30, 2017 12:04 AM
To: Anthony John Souza <ajsouza@mail.sfsu.edu>; Manuel <jdceren@mail.sfsu.edu>
Cc: Dragutin Petkovic <petkovic@sfsu.edu>
Subject: Re: CSC 648-848 Summer Section 01 Team 04 Milestone 3 Meeting

Hi,

Here is our site so far: sfsuse.com/~su17g04/

We've made a lot of improvements, but there's still some functionality we're working on. Registering/logging in/uploading/browsing/seeing your products should all work. I know there's a few bugs, just wanted to get this sent to you for reviewal. Please let me know if you need anything else from us.

Thanks,

Stephen Josey

From: Anthony John Souza <ajsouza@mail.sfsu.edu>
Sent: Wednesday, July 26, 2017 11:07:54 PM
To: Manuel; Stephen Levi Josey
Cc: Dragutin Petkovic
Subject: CSC 648-848 Summer Section 01 Team 04 Milestone 3 Meeting

Hello Team 04,

Today was a good meeting.
The following items were noted during the meeting
- Login doesn't use DB
- Search doesn't use DB (different from VP search)
- Consistent thumbnail size
- Need to move code to server
- Need to connect front-end and bend-end together

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

The purpose of this checkpoint was
- Connecting front-end and back-end

Please reply to this email when sending the checkpoint do not send from a different email.
Please send GitHub link and link to website when replying to email.

Best,
Anthony J Souza

SW Engineering CSC648/848 Section 01 Summer 2017

"PixSale"

Local Team
Team Number 04

Manuel Duran jdceren@mail.sfsu.edu

Stephen Josey SFSU Student

Ryan Jung SFSU Student

Youssef Hakkou SFSU Student

Jeremy Tan SFSU Student

Yoezhou Yu SFSU Student

Milestone 4

08/05/2017

Revision History

Version	Action	Approval	Date
1.0	Creation of original Milestone 4 document for submission	Team Lead/Editor	08/04/2017
2.0	Updated Milestone 4 with professor's feedback	Team Lead/Editor	08/07/2017

Content and structure for Milestone 4 document for review:

Product summary

PixSale

1. Guests can browse and search through media without being registered users.
2. Users can browse through media by category.
3. Users are able to register and login to their own accounts.
4. Passwords are encrypted on the database.
5. All user information is stored on the database.
6. Registered users are able to upload photos to the website.
7. Registered users are able to make purchases on the website.
8. Admins are able to remove any media from the database.
9. Admins are able to remove any user from the database.

URL of the system to be tested: http://sfsuse.com/~su17g04/

Usability Test Plan

1. Test Objectives

Finding images fast and easy is an important aspect of PixSale. The following tests focus on finding images that meets exactly what potential customers are looking for and whether or not this function is easy to use. Users can search for photos and videos by typing the category they are looking for on the search bar. Users can also search images with the search engine by just filtering and selecting the category of their choice to find the the images or videos they want.

2. Test Plan

- *System Setup:* Computer that is running a recent version of Safari, Chrome and Firefox.

- *Starting Point:* Homepage of PixSale.

- *Task to be Accomplished:*

    1. Use search engine to find images or videos in database
       a. Find images of animals
       b. Find images of cars

    2. Select images results using categories
       a. Find an images of flowers
       b. Find an image of a car

- *Intended User:* Registered user, SFSU student.

- *Completion Criteria:* User searched or selected an images category and found a photo or video that met their criteria. The returned values are accurate and satisfactory. As a result, users are able to search and find images or a website response at each step.

- *URL of the system to be tested:* http://sfsuse.com/~su17g04/

Questionnaire

Please circle the answer that best describes your feelings to the following statements.

***Question 1**: It was easy to navigate through the homepage to find images.*

Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree

Comments:

***Question 2**: The returned search engine results were accurate/useful.*

Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree

Comments:

***Question 3**: The filters by category on the site were convenient to find an images/videos.*

Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree

Comments:

***Question 4**: I would recommend PixSale to friends, family members, and other SFSU students.*

Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree

Comments:

QA test plan

Test Objectives:

- To make sure the search functionality returns the correct items

HW and SW setup:

- PixSale does support all modern browser: Chrome, Firefox and Safari. Ensure you have an internet connection and a computer that supports modern browsers.

Feature to be tested:

- Test the search function and filters for each category on all browsers. Make sure to test the functionality, stability and usability. The search feature should return the specific results that were either selected from the category drop-down are typed in the search bar by the user. If the user searches for elephant in the search bar and selects animals from the dropdown category, then it should return an elephant. This category drop-down menu will help narrow the search results.

Test No.	Description	Test input	Expected Output	Pass/Fail
1)	Open the webpage at http://sfsuse.com/~su17g04/	Search: nothing Category: nothing Filter: nothing	Should display all images	PASS Chrome PASS Firefox
2)	Use the category filter and select flowers from the dropdown menu	Search: nothing Category: Flowers Filter: nothing	Should show 1 result of a picture of flower	PASS Chrome PASS Firefox
3)	Use search bar and type random set of numbers	Search: “124132” Category: nothing Filter: nothing	Should display 0 search results	PASS Chrome PASS Firefox
4)	Use the media type filter and select images from the filter	Search: nothing Category: nothing Filter: images	Should display all 7 images	PASS Chrome PASS Firefox
5)	Use search bar and type filter and category.	Search: car Category: cars Filter: images	Should display 3 results of cars	PASS Chrome PASS Firefox

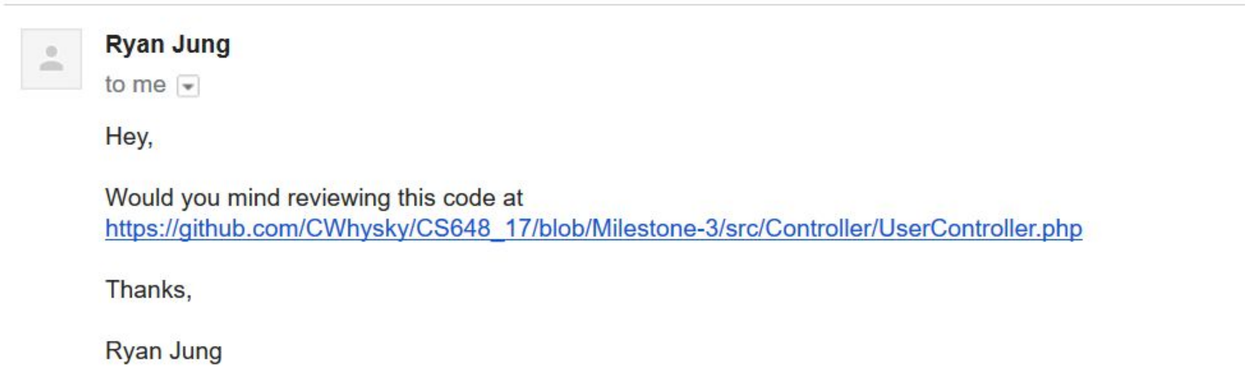
Code Review

By this time you should have chosen a coding style. In the report say what coding style you chose.

Chose the code (substantial portion of it) related to the feature you used for QA and usability test. You need to submit an example of the code (or part of it – 2 pages or so MAX) for this function to be peer reviewed, and document this as follows:

Peer review for AppController (screenshots below):

+-----Peer Review------+
Great use of comments to show what is being handled.
Spaces between if statements and () and {} would be better.
Everything is self-explanatory. I understood what the code was doing by skimming through it.



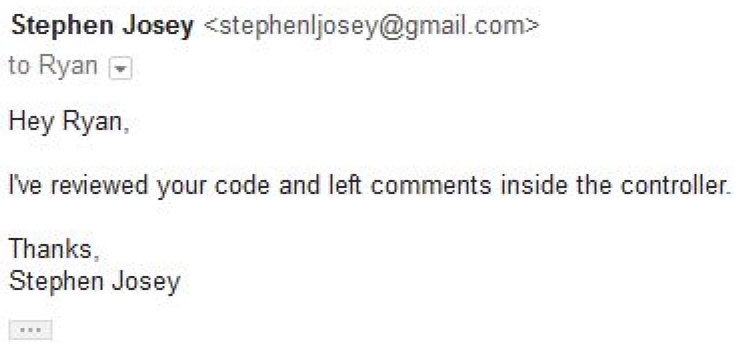
Ryan Jung
to me

Hey,

Would you mind reviewing this code at
https://github.com/CWhysky/CS648_17/blob/Milestone-3/src/Controller/UserController.php

Thanks,

Ryan Jung

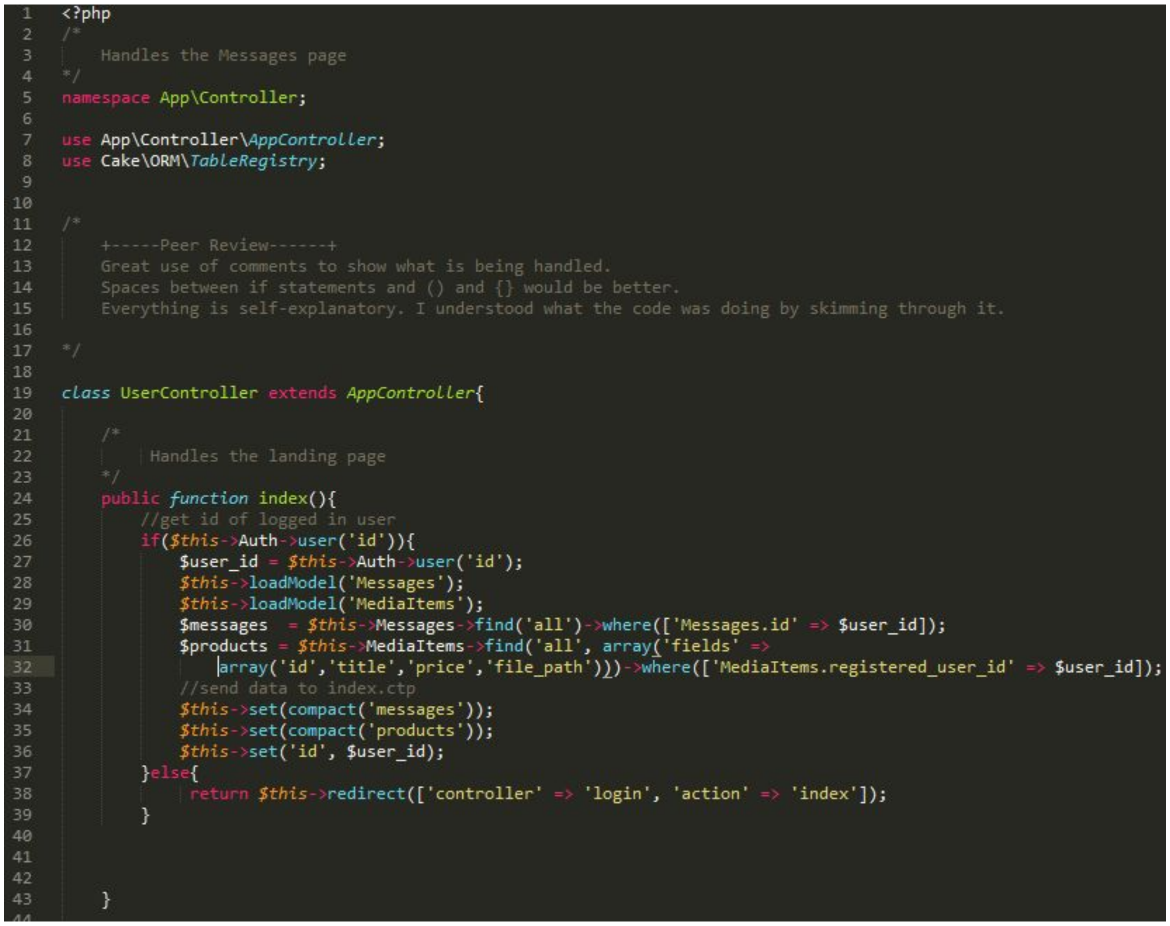


Stephen Josey <stephenljosey@gmail.com>
to Ryan

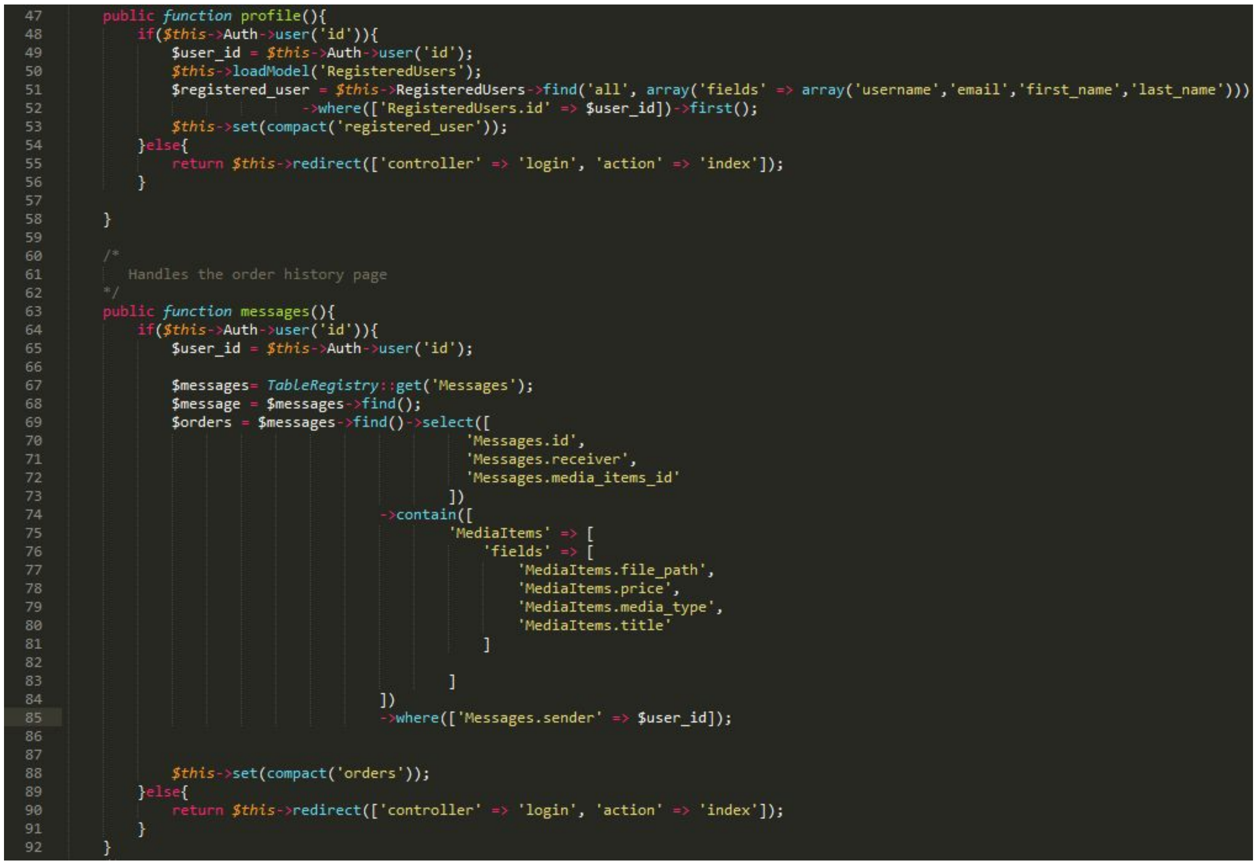
Hey Ryan,

I've reviewed your code and left comments inside the controller.

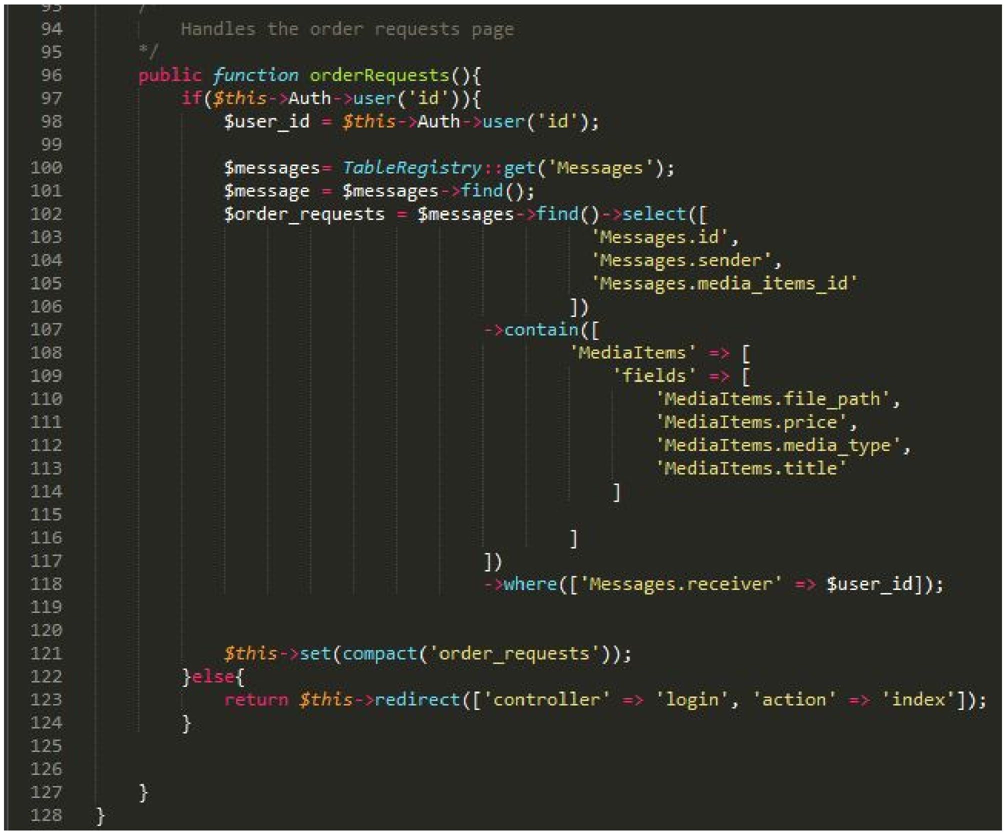
Thanks,
Stephen Josey



```php
<?php
/*
    Handles the Messages page
*/
namespace App\Controller;

use App\Controller\AppController;
use Cake\ORM\TableRegistry;


/*
    +-----Peer Review------+
    Great use of comments to show what is being handled.
    Spaces between if statements and () and {} would be better.
    Everything is self-explanatory. I understood what the code was doing by skimming through it.

*/

class UserController extends AppController{

    /*
        Handles the landing page
    */
    public function index(){
        //get id of logged in user
        if($this->Auth->user('id')){
            $user_id = $this->Auth->user('id');
            $this->loadModel('Messages');
            $this->loadModel('MediaItems');
            $messages  = $this->Messages->find('all')->where(['Messages.id' => $user_id]);
            $products = $this->MediaItems->find('all', array('fields' =>
                array('id','title','price','file_path')))->where(['MediaItems.registered_user_id' => $user_id]);
            //send data to index.ctp
            $this->set(compact('messages'));
            $this->set(compact('products'));
            $this->set('id', $user_id);
        }else{
            return $this->redirect(['controller' => 'login', 'action' => 'index']);
        }



    }
```



```php
    public function profile(){
        if($this->Auth->user('id')){
            $user_id = $this->Auth->user('id');
            $this->loadModel('RegisteredUsers');
            $registered_user = $this->RegisteredUsers->find('all', array('fields' => array('username','email','first_name','last_name')))
                                    ->where(['RegisteredUsers.id' => $user_id])->first();
            $this->set(compact('registered_user'));
        }else{
            return $this->redirect(['controller' => 'login', 'action' => 'index']);
        }

    }

    /*
       Handles the order history page
    */
    public function messages(){
        if($this->Auth->user('id')){
            $user_id = $this->Auth->user('id');

            $messages= TableRegistry::get('Messages');
            $message = $messages->find();
            $orders = $messages->find()->select([
                                        'Messages.id',
                                        'Messages.receiver',
                                        'Messages.media_items_id'
                                       ])
                                ->contain([
                                       'MediaItems' => [
                                           'fields' => [
                                               'MediaItems.file_path',
                                               'MediaItems.price',
                                               'MediaItems.media_type',
                                               'MediaItems.title'
                                           ]

                                       ]
                                ])
                                ->where(['Messages.sender' => $user_id]);


            $this->set(compact('orders'));
        }else{
            return $this->redirect(['controller' => 'login', 'action' => 'index']);
        }
    }
```



```php
       Handles the order requests page
    */
    public function orderRequests(){
        if($this->Auth->user('id')){
            $user_id = $this->Auth->user('id');

            $messages= TableRegistry::get('Messages');
            $message = $messages->find();
            $order_requests = $messages->find()->select([
                                        'Messages.id',
                                        'Messages.sender',
                                        'Messages.media_items_id'
                                       ])
                                ->contain([
                                       'MediaItems' => [
                                           'fields' => [
                                               'MediaItems.file_path',
                                               'MediaItems.price',
                                               'MediaItems.media_type',
                                               'MediaItems.title'
                                           ]

                                       ]
                                ])
                                ->where(['Messages.receiver' => $user_id]);


            $this->set(compact('order_requests'));
        }else{
            return $this->redirect(['controller' => 'login', 'action' => 'index']);
        }


    }
}
```

Self-check on best practices for security

- **Assets to be protected**:

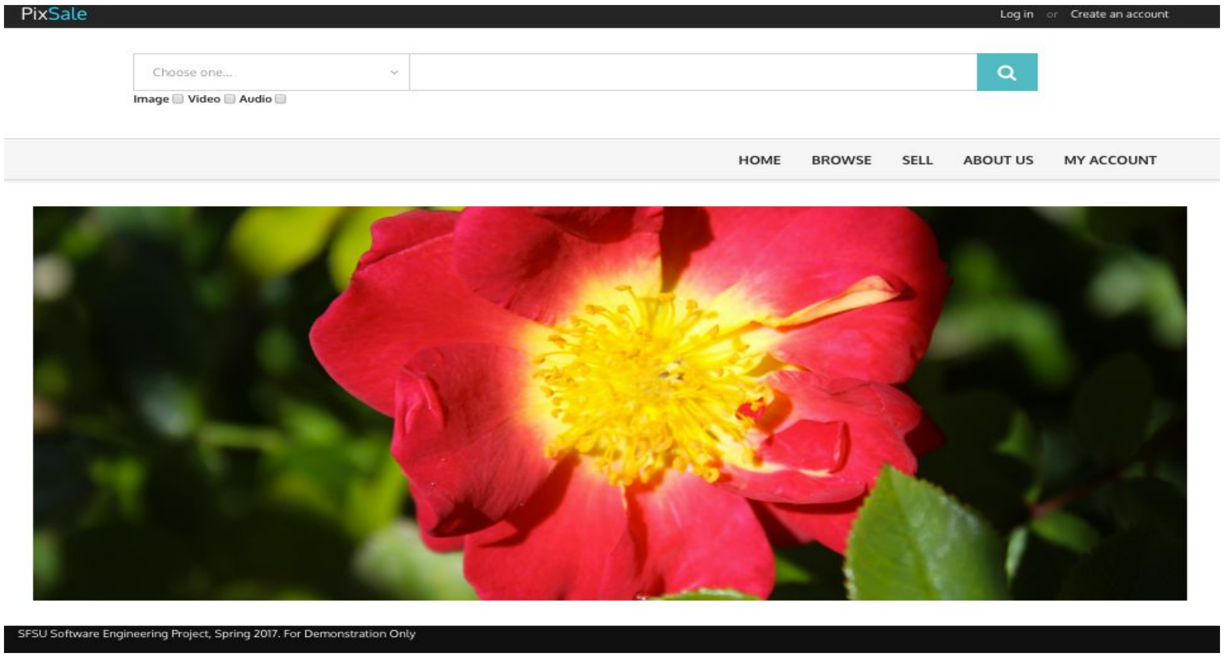
    1) Registered User's account
    2) Registered User's uploaded media items.

- Password are encrypted on the database using CakePHP's hash function

- CakePHP's find() method prepares SQL statements to prevent SQL injection from occurring. We use CakePHP's find() method with the user's input from the search bar and let CakePHP handle preparing the SQL statements. We follow CakePHP guidelines for database queries and that protects us from most SQL injections. The search bar will also check for string text not to exceed 30 alphanumeric characters. The search bar will also check for empty text in the search bar as well as any unselected category or media type. If no filters are selected then the search will display everything.

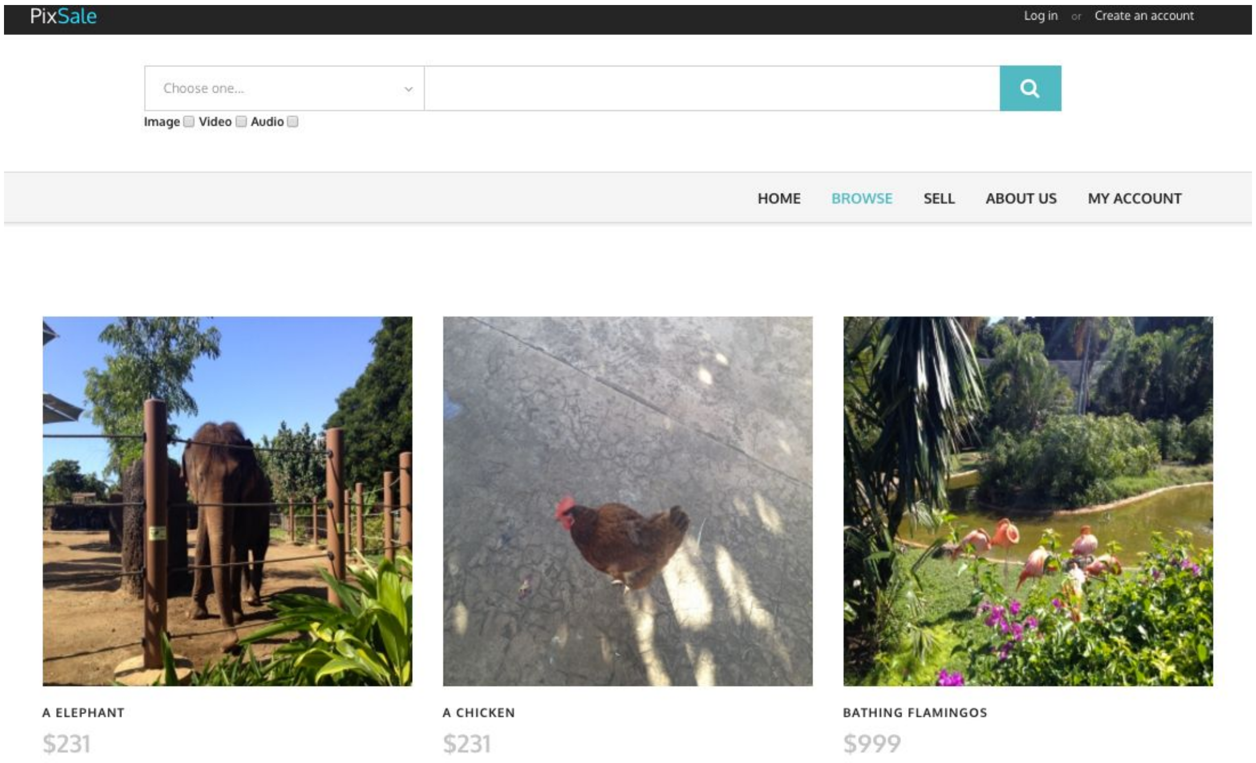
Screen Shots of final product

HOME


PixSale
Log in or Create an account
Choose one...
Image Video Audio
HOME BROWSE SELL ABOUT US MY ACCOUNT

SFSU Software Engineering Project, Spring 2017. For Demonstration Only

Browse

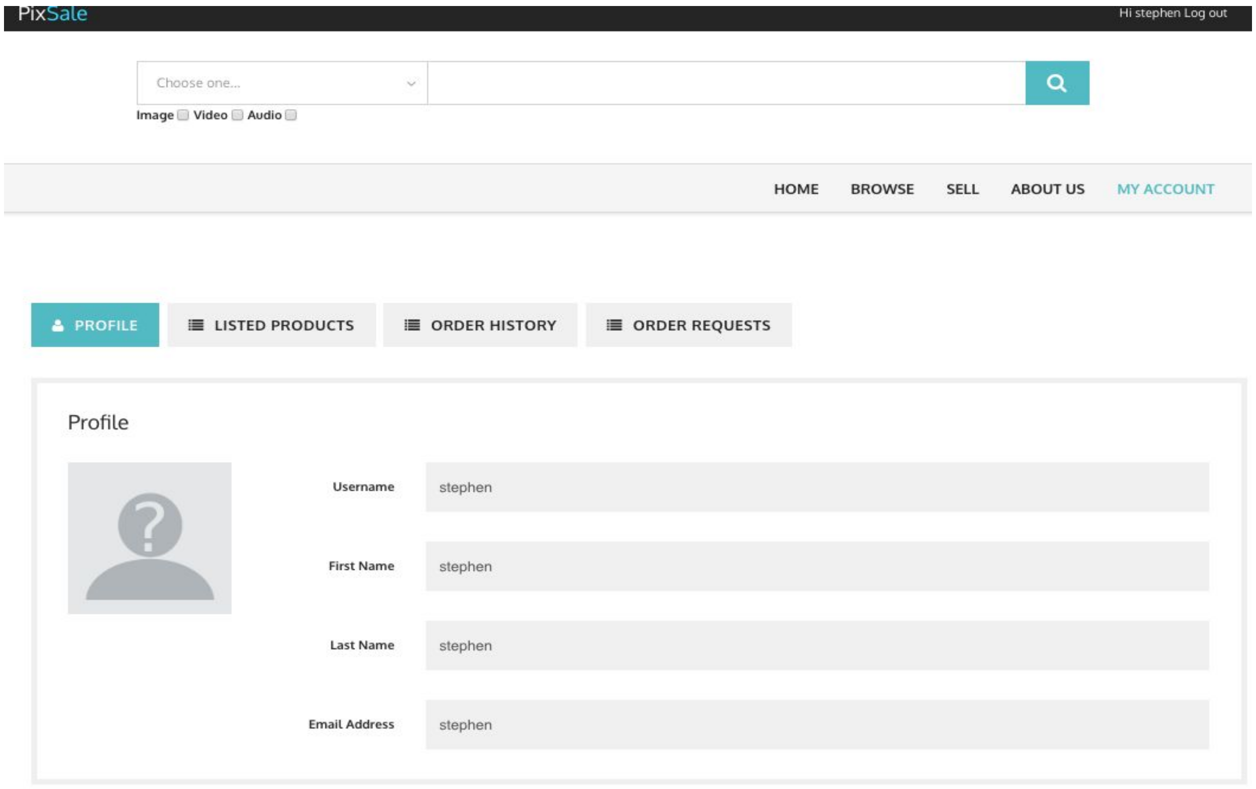

PixSale
Log in or Create an account
Choose one...
Image Video Audio
HOME BROWSE SELL ABOUT US MY ACCOUNT

A ELEPHANT
$231

A CHICKEN
$231

BATHING FLAMINGOS
$999

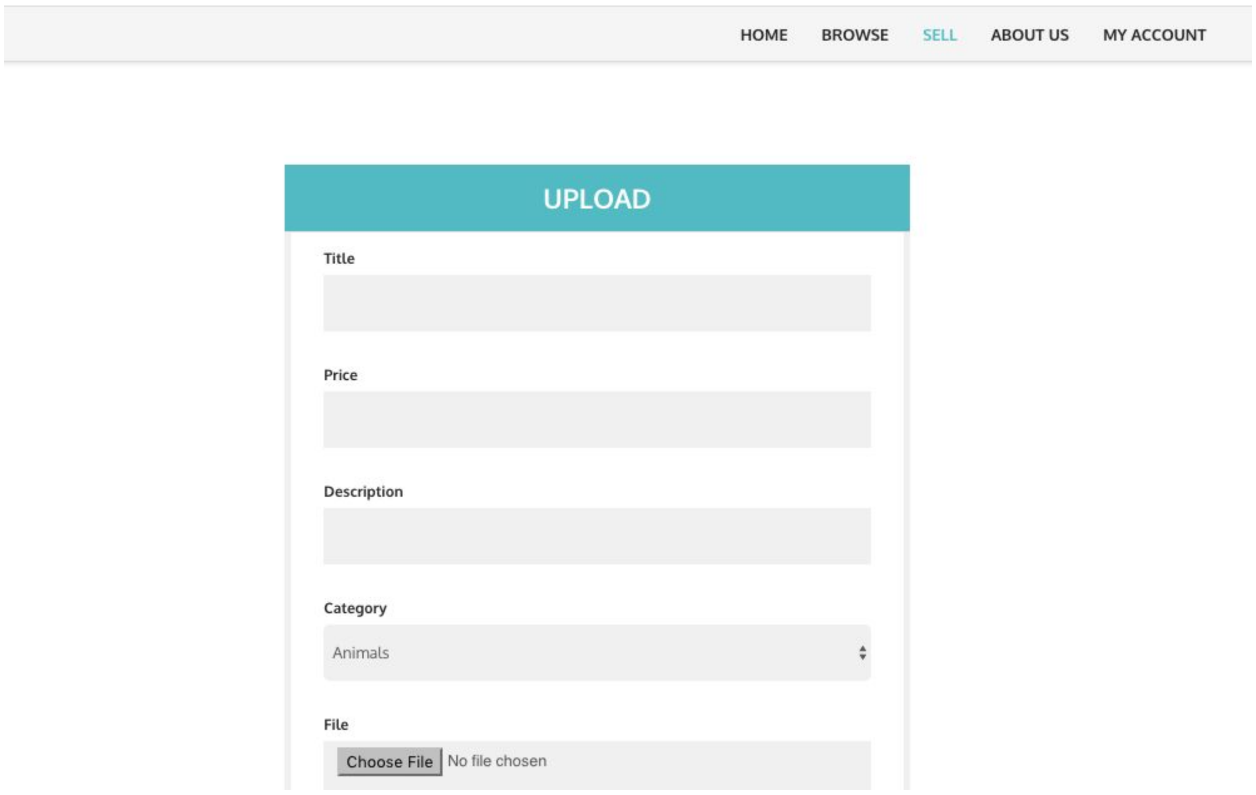
Profile



Profile

Username	stephen
First Name	stephen
Last Name	stephen
Email Address	stephen

Upload



Title

Price

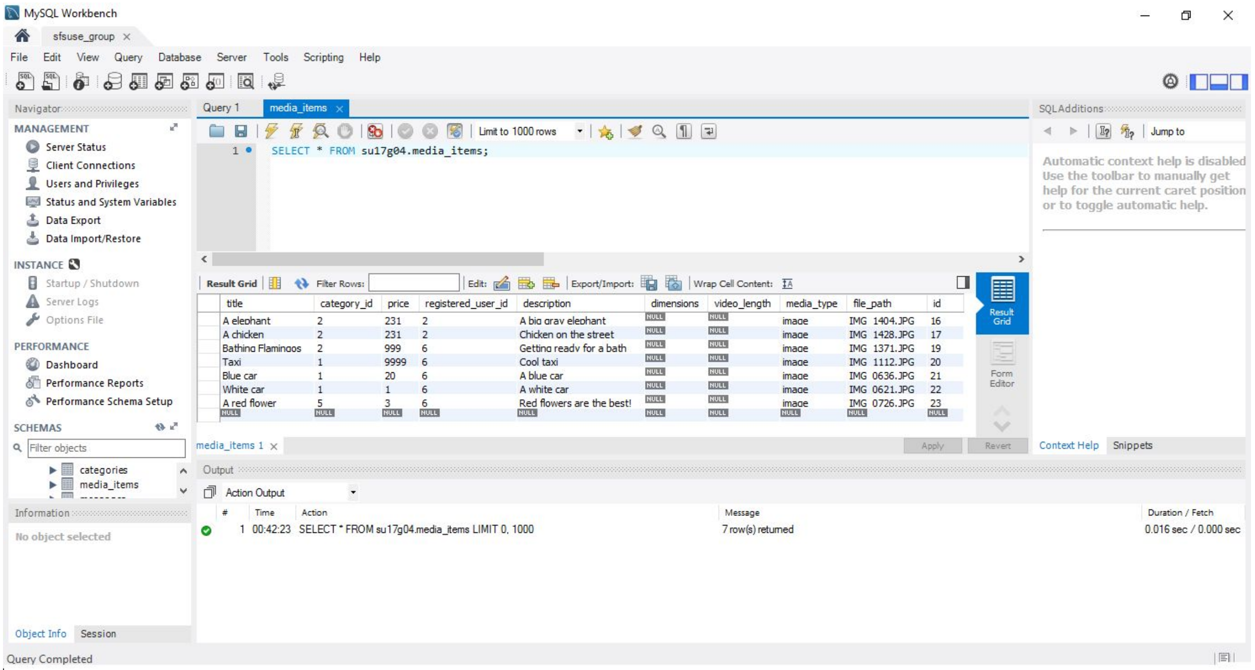
Description

Category

Animals

File

Workbench screen shots of Database

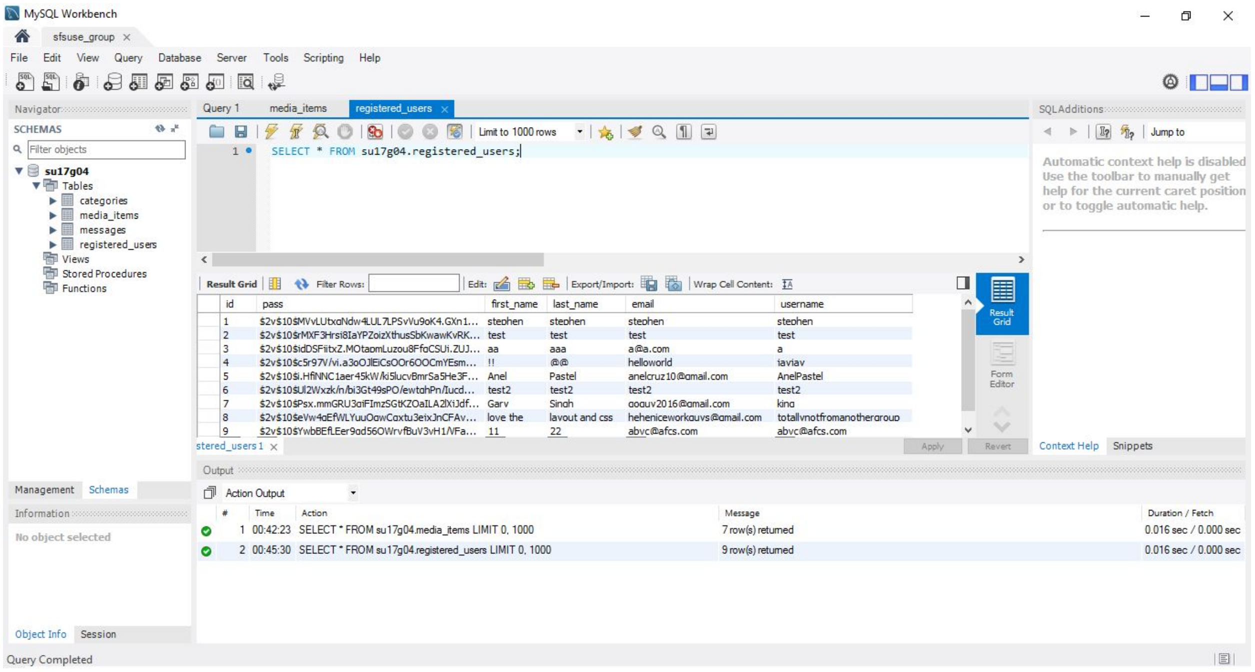


```
SELECT * FROM su17g04.media_items;
```

title	category_id	price	registered_user_id	description	dimensions	video_length	media_type	file_path	id
A elephant	2	231	2	A big gray elephant	NULL	NULL	image	IMG_1404.JPG	16
A chicken	2	231	2	Chicken on the street	NULL	NULL	image	IMG_1428.JPG	17
Bathing Flamingos	2	999	6	Getting ready for a bath	NULL	NULL	image	IMG_1371.JPG	19
Taxi	1	9999	6	Cool taxi	NULL	NULL	image	IMG_1112.JPG	20
Blue car	1	20	6	A blue car	NULL	NULL	image	IMG_0636.JPG	21
White car	1	1	6	A white car	NULL	NULL	image	IMG_0621.JPG	22
A red flower	5	3	6	Red flowers are the best!	NULL	NULL	image	IMG_0726.JPG	23
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

```
#  Time      Action                                          Message           Duration / Fetch
1  00:42:23  SELECT * FROM su17g04.media_items LIMIT 0, 1000  7 row(s) returned  0.016 sec / 0.000 sec
```

Query Completed



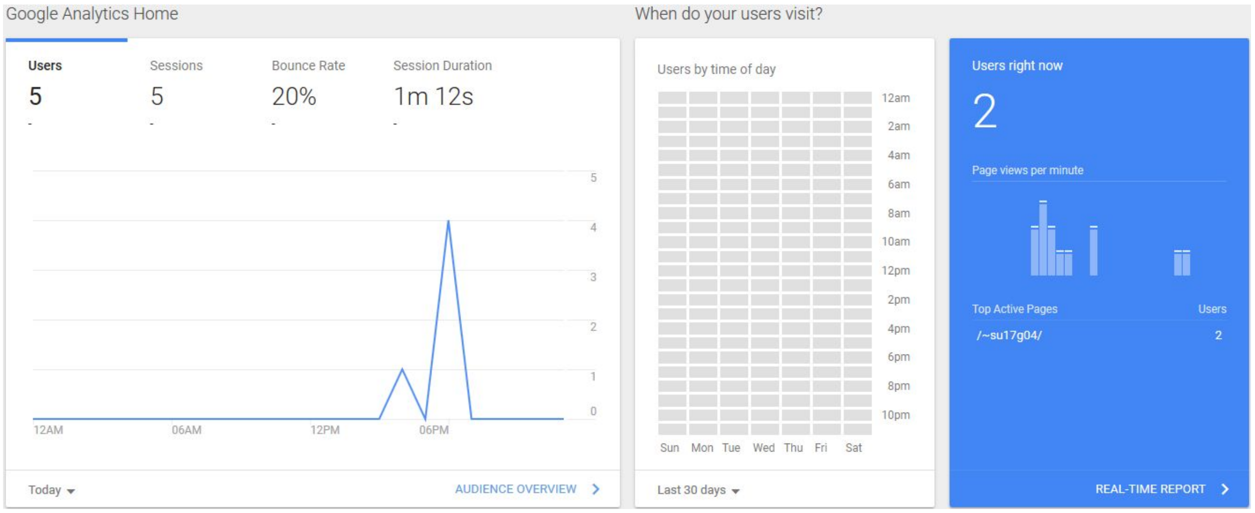
```
SELECT * FROM su17g04.registered_users;
```

id	pass	first_name	last_name	email	username
1	$2y$10$MVvLUtxqNdw4LUL7LPSvVu9oK4.GXn1...	stephen	stephen	stephen	stephen
2	$2y$10$rMXF3Hrsi8IaYPZoizXthusSbKwawKvRK...	test	test	test	test
3	$2y$10$idDSFiitxZ.MOtapmLuzou8FfqCSUi.ZUJ...	aa	aaa	a@a.com	a
4	$2y$10$c5r97V/vi.a3oOJlEiCsOOr6OOCmYEsm...	!!	@@	helloworld	jayjay
5	$2y$10$i.HfiNNC1aer45kW/ki5lucvBmrSa5He3F...	Anel	Pastel	anelcruz10@gmail.com	AnelPastel
6	$2y$10$Ul2Wxzk/n/bi3Gt49sPO/ewtqhPn/Iucd...	test2	test2	test2	test2
7	$2y$10$Psx.mmGRU3qiFImzSGtKZOaILA2lXiJdf...	Gary	Singh	goguy2016@gmail.com	king
8	$2y$10$eVw4qEfWLYuuOqwCqxtu3eixJnCFAv...	love the	layout and css	heheniceworkguys@gmail.com	totallynotfromanothergroup
9	$2y$10$YwbBEfLEer9qd56OWrvfBuV3vH1/VFa...	11	22	abyc@afcs.com	abyc@afcs.com

```
#  Time      Action                                               Message           Duration / Fetch
1  00:42:23  SELECT * FROM su17g04.media_items LIMIT 0, 1000       7 row(s) returned  0.016 sec / 0.000 sec
2  00:45:30  SELECT * FROM su17g04.registered_users LIMIT 0, 1000  9 row(s) returned  0.016 sec / 0.000 sec
```

Query Completed

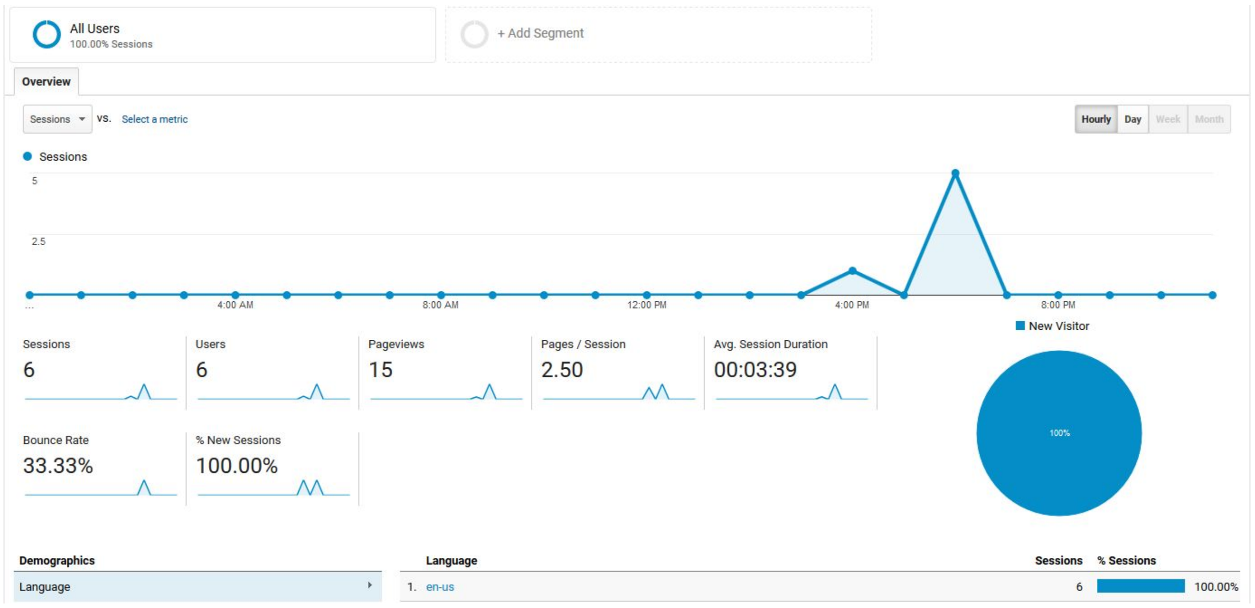
Google analytics



Users 5 · Sessions 5 · Bounce Rate 20% · Session Duration 1m 12s

Users right now 2

Top Active Pages	Users
/~su17g04/	2



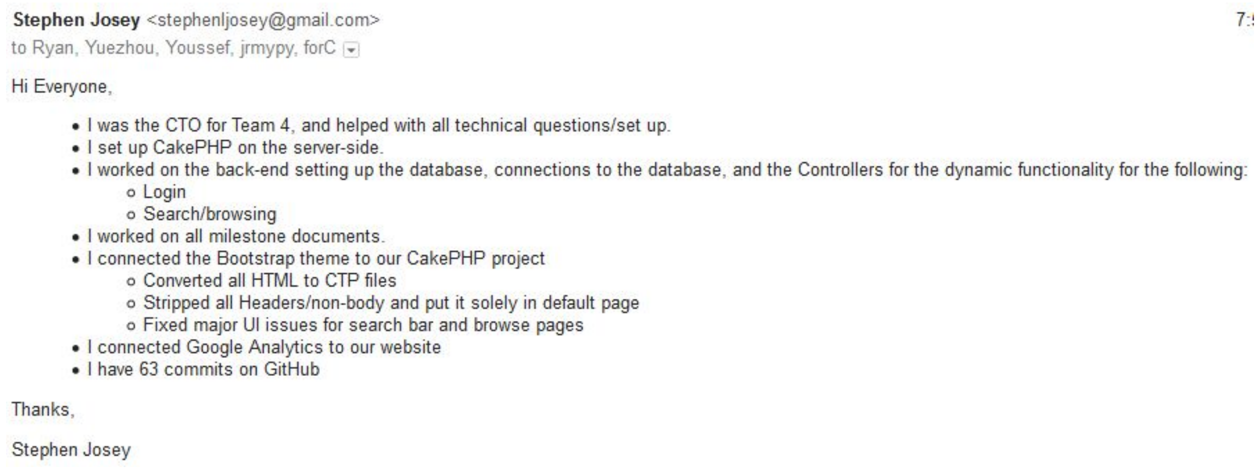
Sessions	Users	Pageviews	Pages / Session	Avg. Session Duration
6	6	15	2.50	00:03:39

Bounce Rate	% New Sessions
33.33%	100.00%

Demographics	Language	Sessions	% Sessions
Language	1. en-us	6	100.00%

Team member contributions

Stephen Josey's email:



Stephen Josey <stephenljosey@gmail.com> 7:

to Ryan, Yuezhou, Youssef, jrmypy, forC

Hi Everyone,

- I was the CTO for Team 4, and helped with all technical questions/set up.
- I set up CakePHP on the server-side.
- I worked on the back-end setting up the database, connections to the database, and the Controllers for the dynamic functionality for the following:
  - Login
  - Search/browsing
- I worked on all milestone documents.
- I connected the Bootstrap theme to our CakePHP project
  - Converted all HTML to CTP files
  - Stripped all Headers/non-body and put it solely in default page
  - Fixed major UI issues for search bar and browse pages
- I connected Google Analytics to our website
- I have 63 commits on GitHub

Thanks,

Stephen Josey

Main Challenges and Solutions

Some of the main challenges we faced were determining a meeting times and determining work delegation base on each student’s skills. As the class progress and the group got to know each other, we began to work on the tasks based on each person’s ability. One idea to asses early on each person’s skills would be to have a non related work meeting early on. Meeting was also complicated at times. Next time I would use a softwer form such as Doodle.com to determinate early on people’s time availability to meetup.

We also used the school resources such as reserving rooms in the library which help organize our meetings but we did so late in the project as well. The use of resources early on would be something I would do in the future to get more productivity.