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.

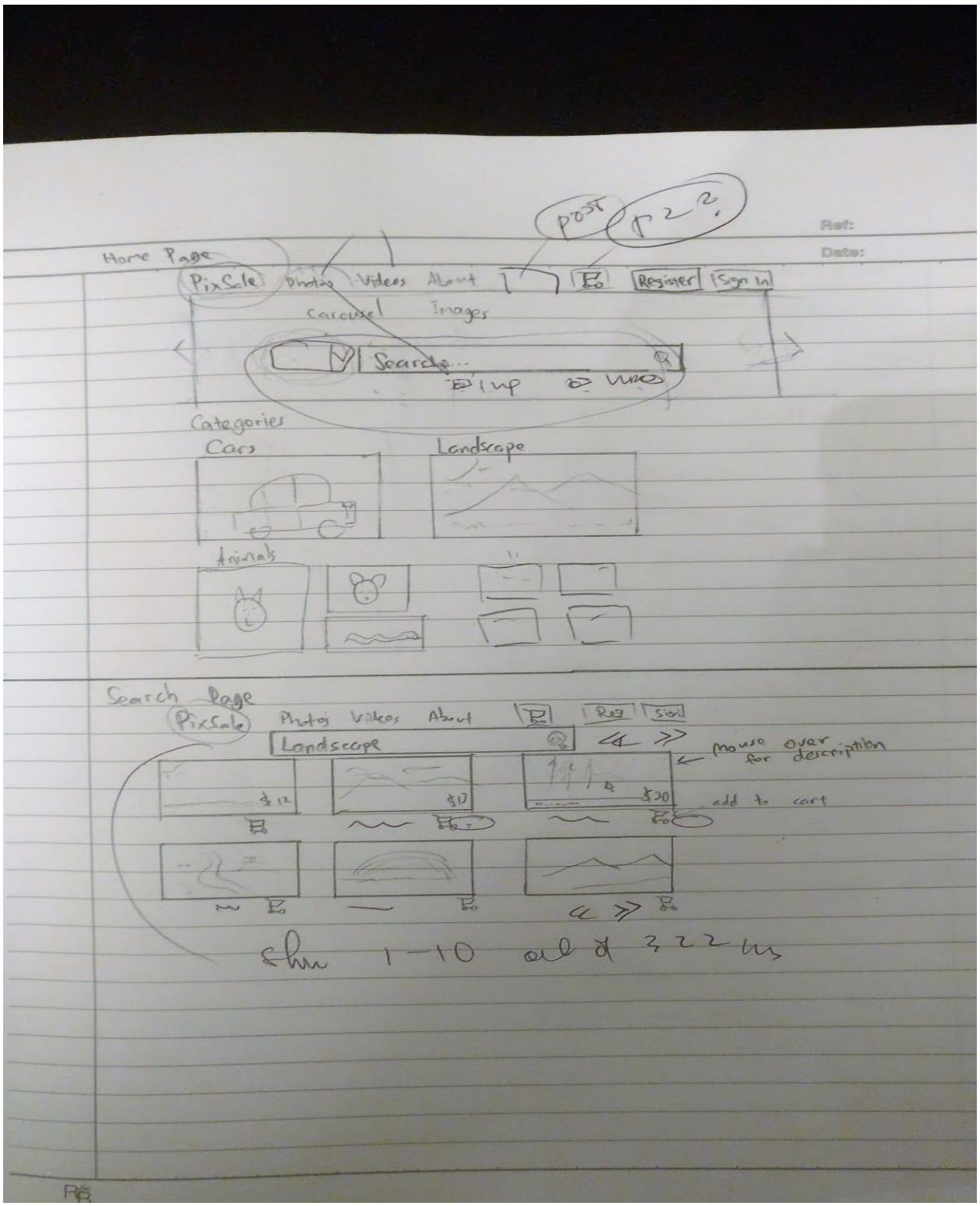
**Priority 2 requirements:**

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).

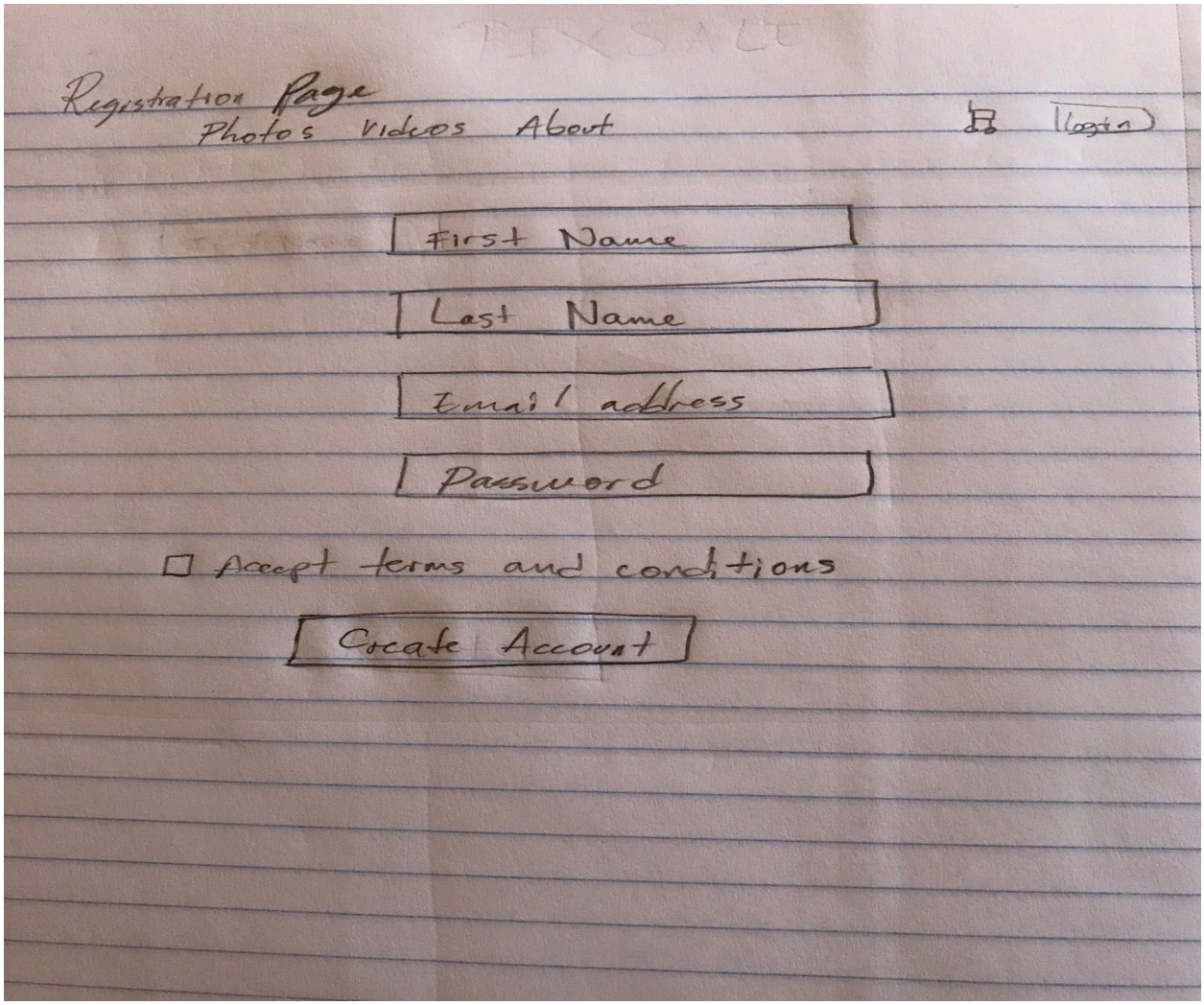
# 5. UI Mockups and Storyboards (high level only)



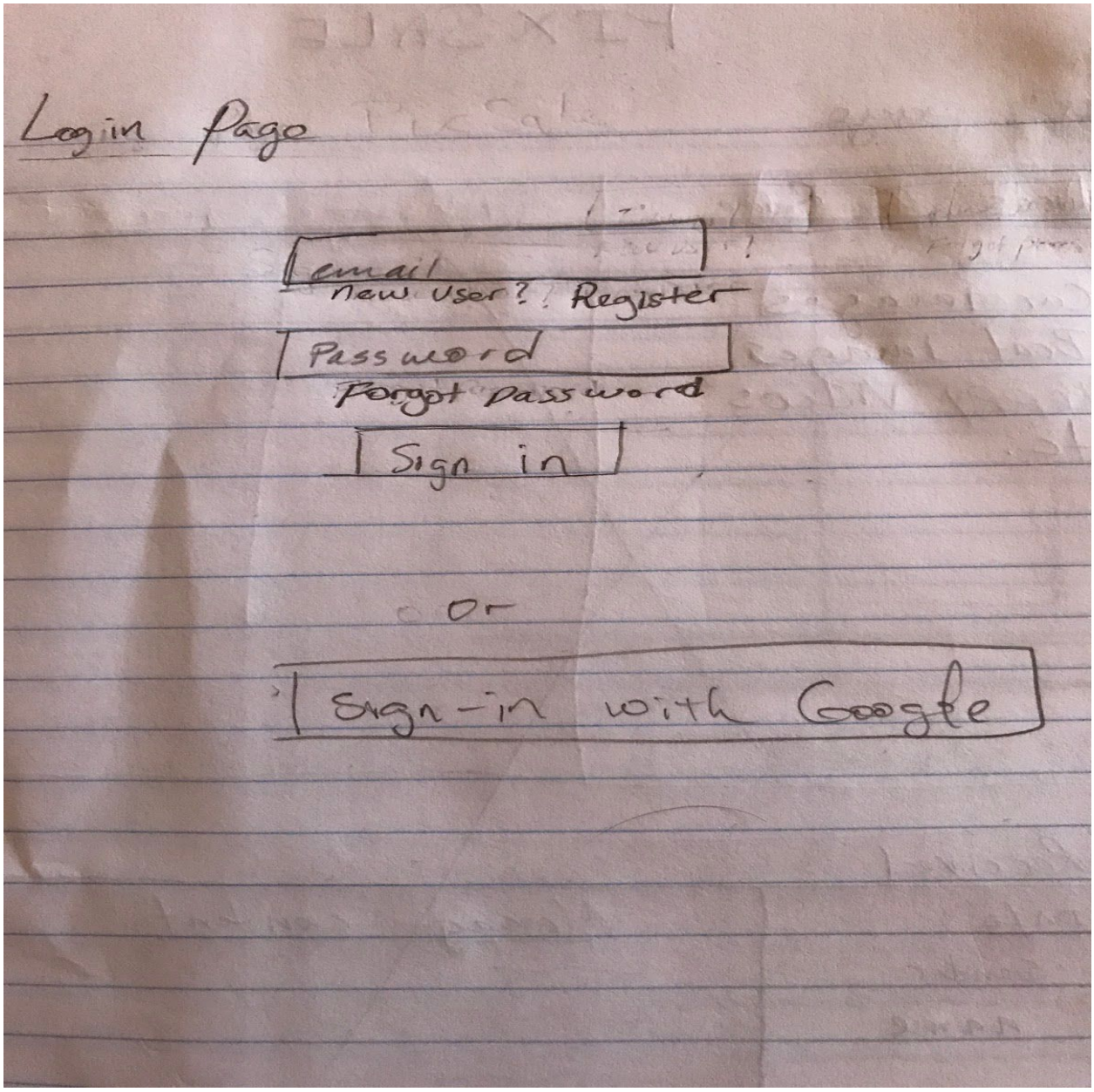
The **guest** will begin at the Home Page where he is able to view previews of **media** from different categories. The guest will be able to filter the search for images or videos by using checking checkboxes near the search bar. The guest can also choose any of the available categories in a drop down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

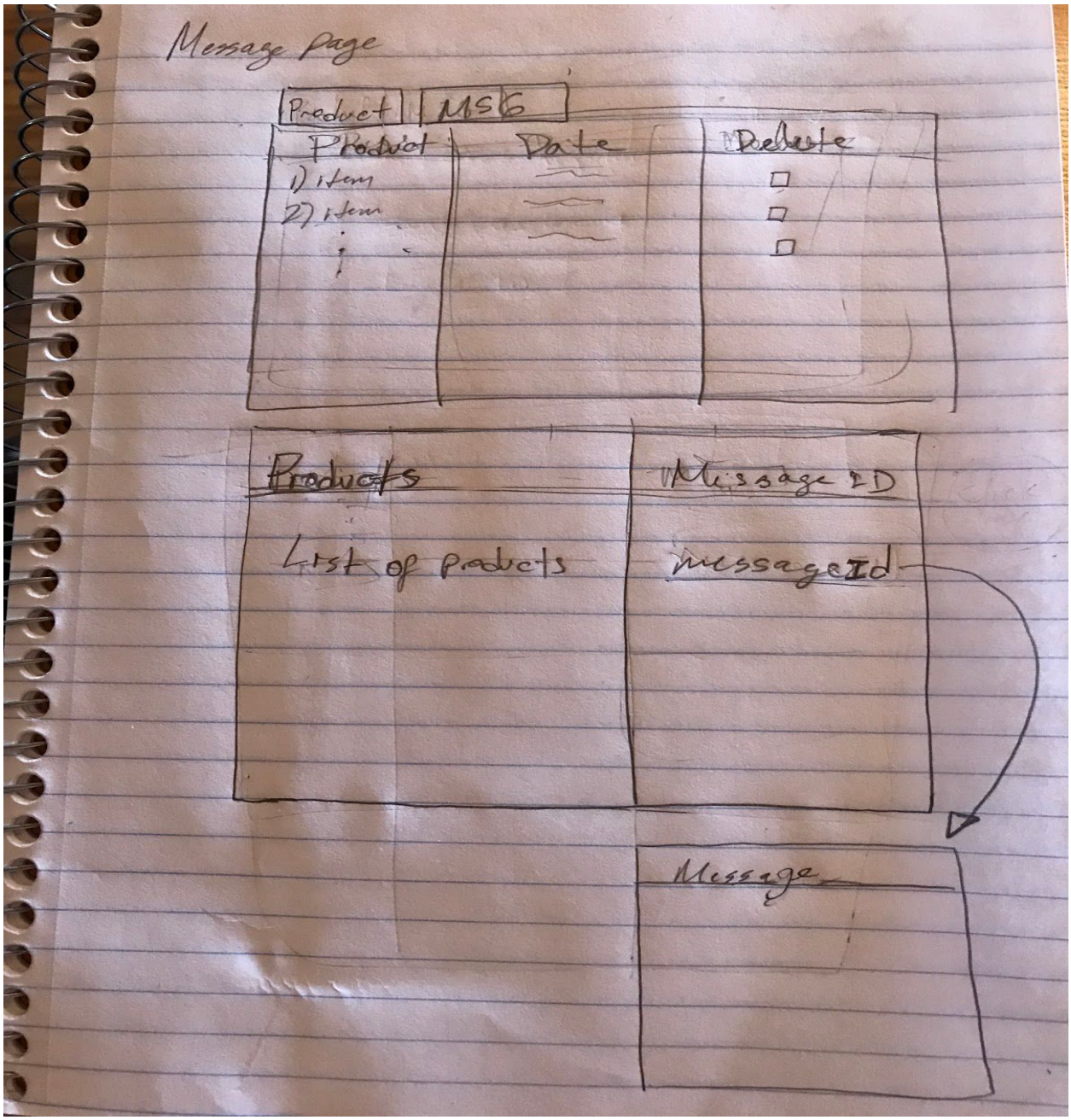
The search page will also display how many media items are currently being displayed as well as how many search results matched the **guest's** query. The **guest** will only be display small amounts of media items at a time and will be able to browse through more media items using navigation arrows.




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:

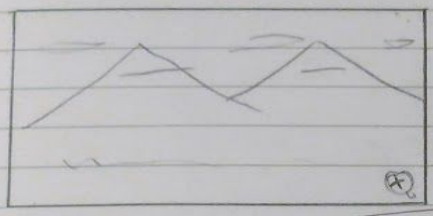
Date:

Photo Page

number of items in cart

PixSale  Photos  Videos  About  1  Register  Signin

Search



Blue Hills - Stock Image

| | Resolution | Size | Price | |
|---|---|---|---|---|
| ○ | S | 500x500 | $ | |
| ● | M | 1000 x 1000 | $ | ← selected |
| ○ | L | 2000 x 2000 | $ | |

Add to cart

tags :

image ID : 2434

copyright : ...

Related Images

< >

Related keywords

... , ... , ...

Cart Page

PixSale  Photos  Videos  About  Register  Signin

Search

remove

X

X

DIFF. selleu?

subtotal : $ 300.00 USD

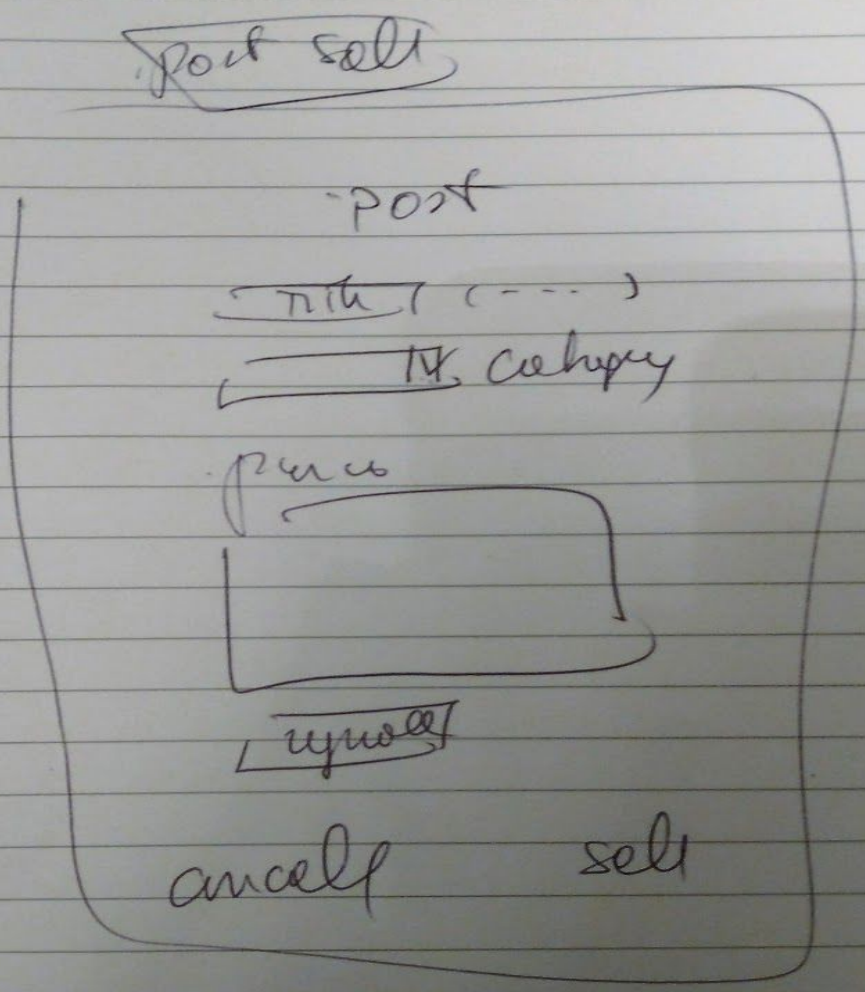
Continue with Purchase

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:



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

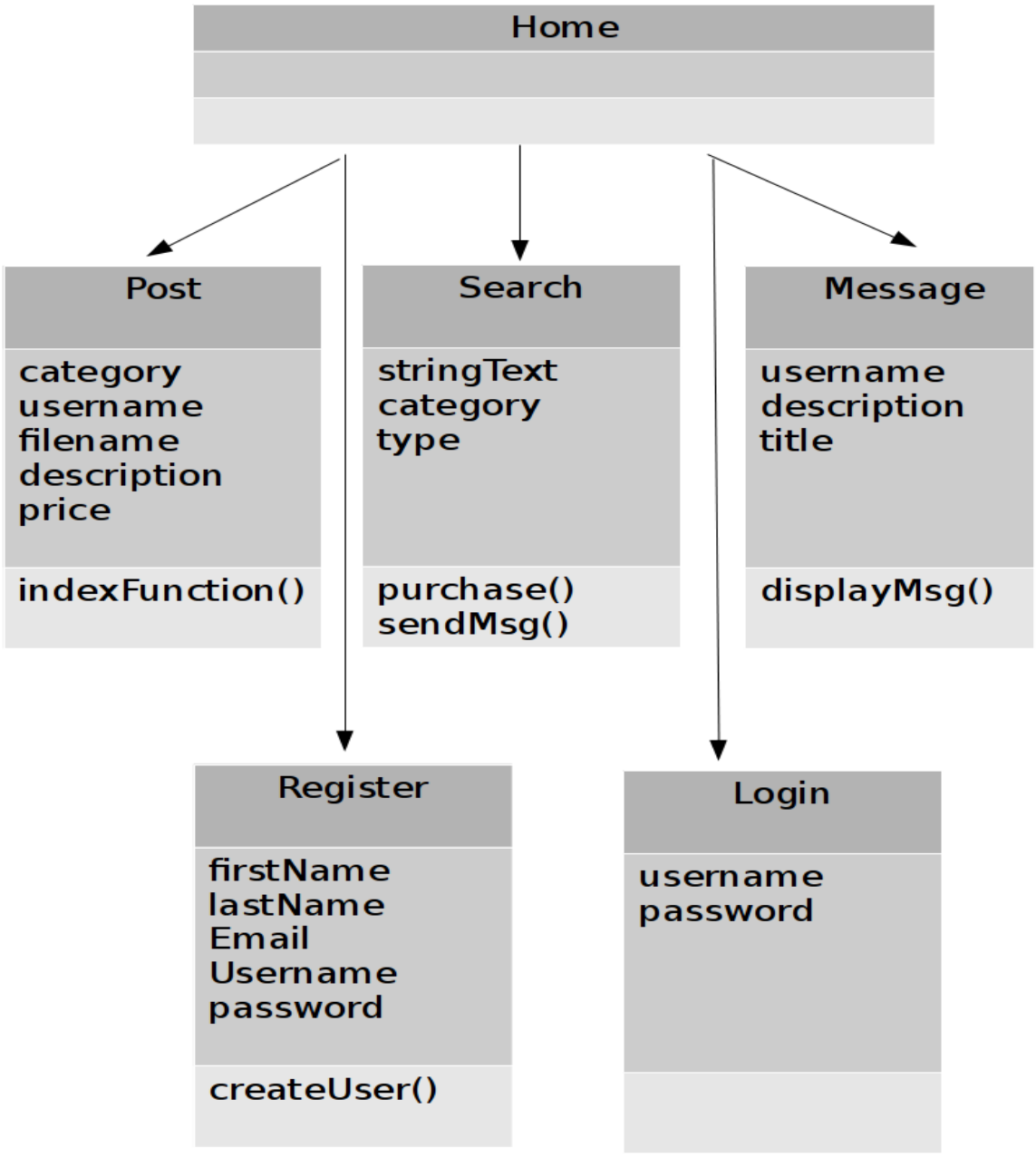
*Media items* - title , description , price, category, user id , media type, file path, video length, image size

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.

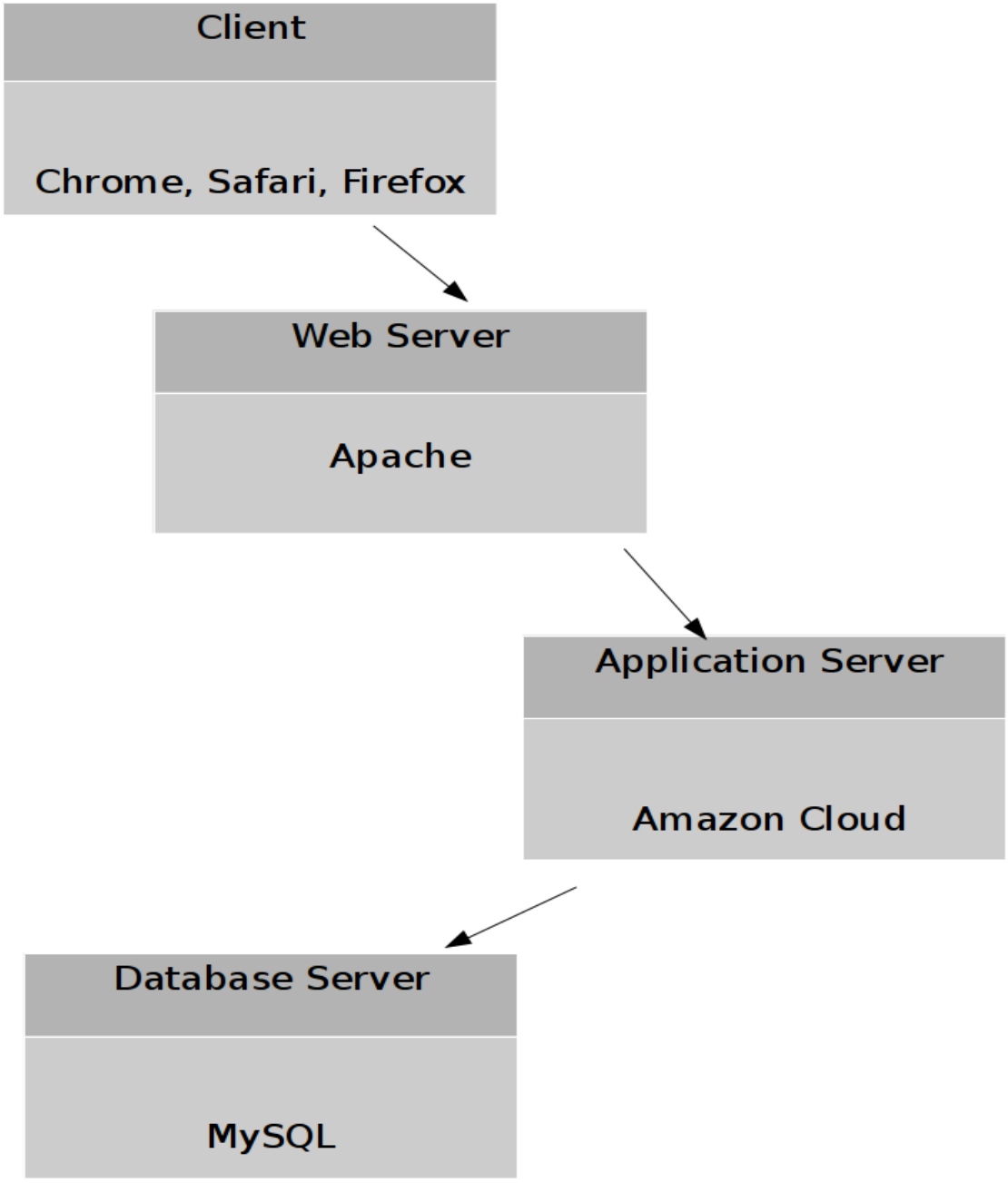
To search for media items we will search against the category, title, and media type in *Media items*. The user will select a category and media type and can choose to enter a key term in the search bar. The search will then query *Media Items* table to find any records whose title contains the user's 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.