SW Engineering
CSC648/848 Section 02
Spring 2017

Gator Trader

Local Team 08

Rebecca Stankus, SFSU
rstankus@mail.sfsu.edu

Farbod Zolghadri, SFSU
Kai Kaur, SFSU
Kyle Kamibayashi, SFSU
Krunal Shah, SFSU
Jeremy Ibay, SFSU

Milestone 2

March 16, 2017

History Table		
Date	**Revision**	**Done**
3/1	Revised from M1	

1. USE CASES V2

GUEST USER

Bob is a student of San Francisco State University. He was curious when he heard about a buy and sell application meant for students who attended SFSU. After Bob opens Gator Trader for the first time, he has the ability to browse through **ITEM**s that are on sale. After seeing an ITEM he wants to purchase, only then is he prompted to login, or register since this would be his first time. Being register allows him access to post and sell ITEMs to his liking. Bob decides he does not want to register yet, and decides that he rather put something up for sale instead. He goes to the sell section of the site and proceeds to sell his I-clicker. Before he is allowed to officially list an ITEM for sale, he is prompted to register, since he still has not registered. This time, Bob decides to register and posts his very first listing.

Registered SELLER

Sarah is a student of San Francisco State University and a regular **SELLER** of ITEMs on Gator Trader. She wants to put her dress on sale. She clicks the sell button and is instructed to add title, description, and up to four pictures among other types of data. Before she posts her dress, she is prompted to sign in verify it's her. Once it's officially posted, she looks through her profile to check her previous postings to see if anyone has bought her ITEMs.

Registered BUYER

John is also a student of San Francisco State University who is a frequent **BUYER** of ITEMs listed on Gator Trader. He frequently checks the site to see if there are any new postings of books for his desired classes. John can single out the selection of ITEMs by changing categories from books to electronics to clothes. So when John is in need of new shoes or chargers for his smart phone, he can easily switch categories with a single button. Before he confirms his purchase, he is prompted to contact the SELLER, and he makes an appointment with the SELLER in the Cesar Chavez building.

ADMINistration

Kyle is an **ADMIN** of Gator Trader. He is responsible for building the application and occasionally logs on to see if there are any bugs or problems with the USERs. As Kyle browses the ITEMs in all the different categories, he notices a few ITEMs that are inappropriate. He proceeds to delete the posts, and notifies the SELLER why the post was deleted. A ban will be given to the SELLERs who post inappropriate listings. Kyle continues to search the site and notices a bug in the selling section that doesn't let the SELLER put up more than 3 pictures, when it clearly says SELLERs are allowed to post 4. He flags the bug to let the developers know where the issue occurred. 30 days after an ITEM is sold, Kyle deletes the ITEM and its attributes.

2. DATA DEFINITIONS V2

USER: Any person who accesses Gator Trader. This includes all GUESTs, SELLERs, BUYERs, and ADMINs. USER's data may be stored in the USERs database.

REGISTERED USER: Any USER who has created an account. A REGISTERED USER will have data stored in the USERs database. This data is collected by a form that will be filled out when the USER creates an account. A REGISTERED USER may buy or sell ITEMs and has a USER id, first name, last name, email, password, screen name, and markers for whether they are a SELLER or ADMIN. The USER will provide all data except id and SELLE/ ADMIN markers. We may include a USER profile picture at a later date that would be uploaded as a <2MB .jpg by the USER.

GUEST: An unregistered USER. A GUEST can view ITEMs for sale, but cannot sell or buy. No data is stored for a GUEST.

SELLER: A registered USER who has posted one or more ITEMs to be sold. A SELLER has all REGISTERED USER attributes and is marked as a SELLER upon posting an item. SELLERs will be able to access any ITEMS they have sold in the last 90 days.

BUYER: A registered USER who is making a purchase of an ITEM. A BUYER has all REGISTERED USER attributes. BUYERs will receive a receipt including all info of ITEM for their records via email upon purchase of an ITEM. After this, the ITEM will no longer be accessible to anyone other than the SELLER.

ADMIN: A registered USER who has exclusive access to the database where ITEMs or USERs can be modified or deleted. The ADMIN has the ability to ban a USER for misconduct and delete USERs and ITEMs. An ADMIN has all REGISTERED USER attributes and is marked as an ADMIN.

ITEM: A categorized object to be sold. An ITEM has, for example, an id, a price, category, SELLER, 4 binary image descriptors to show whether an image exists, description, title, whether the item has been sold, and date/time posted. This info will be collected via form from the SELLER. All ITEM's info except actual IMAGEs is stored in the ITEMs database.

3. FUNCTIONAL SPECS V2

Priority 1

Guest User

- GUESTs shall have access to browse the website.
- GUESTs shall register if they want access to buying and selling.
- GUESTs shall be able to login if they already have their account.
- GUESTs shall switch between categories while browsing.
- GUESTs shall be able to see the About page.
- GUESTs shall have access to see a particular ITEM.
- GUESTs shall be able to search ITEMs from search.

Registered User

- Registered USERs shall have access to all privileges of GUESTs and also the ability to purchase items.
- Registered USERs shall have access to selling items.
- Registered USERs shall be prompted to input pictures for items to sell.
- Registered USERs shall be prompted to input title and description for items to sell.
- Registered USERs shall be able to contact SELLERS in order to arrange a purchase.
- Registered USERs shall be able to see his profile.
- Registered USERs shall be able to logout from his account.
- Registered USERs shall see previous items bought/sold.

Admin

- ADMINs shall take down postings that are deemed inappropriate.
- ADMINs shall be able to edit and change the post.

ITEMS

- ITEMs shall have titles, descriptions, prices, and images.
- ITEMs shall be categorized to enable searching.

Priority 2

Guest User

- GUESTs shall be able to sort items by price.
- GUESTs shall be able to sort items by latest.

Registered User

- Registered USERs shall have access to all privileges of GUESTs .
- Registered USERs shall be able to sort items by price.
- Registered USERs shall be able to sort items by latest.

Admin

- ADMINs shall ban accounts that use the site against school policy.

Priority 3

Guest User

- GUESTs shall put items they might want to buy into the shopping cart.
- GUESTs shall be able to sort items by ratings.

Registered User

- Registered USERs shall be able to sort items by ratings.
- Registered USERs shall be able to rate another Registered USERs.
- Registered USERs shall be able to change the name of ITEM he posted.
- Registered USERs shall be able to change the price of ITEM he posted.
- Registered USERs shall be able to change the description of ITEM he posted.
- Registered USERs shall be able to change the category of ITEM he posted.
- Registered USERs shall be able to delete the image of ITEM he posted.

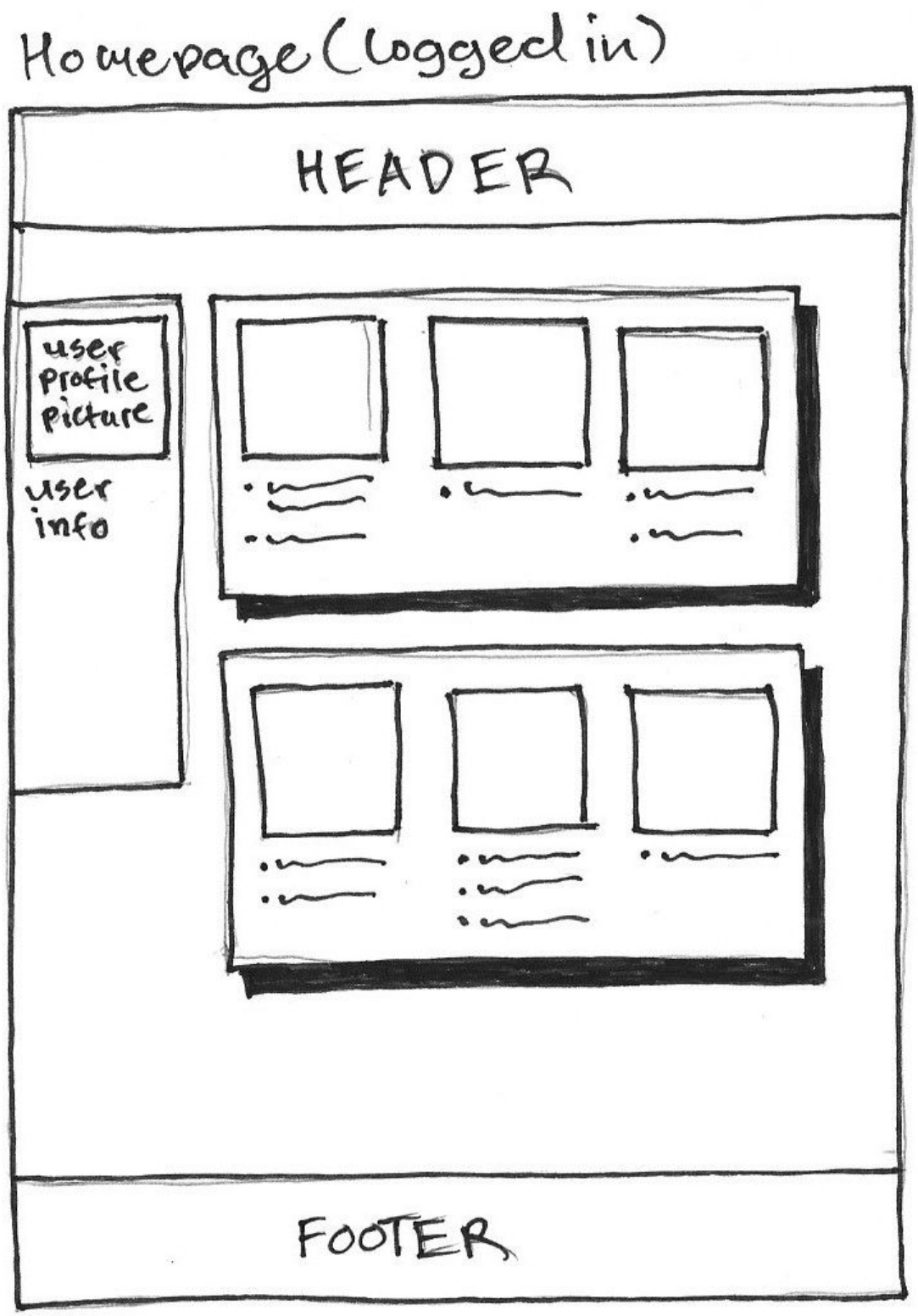
Admin

- ADMINs shall be able to send warnings to the Registered USERs.

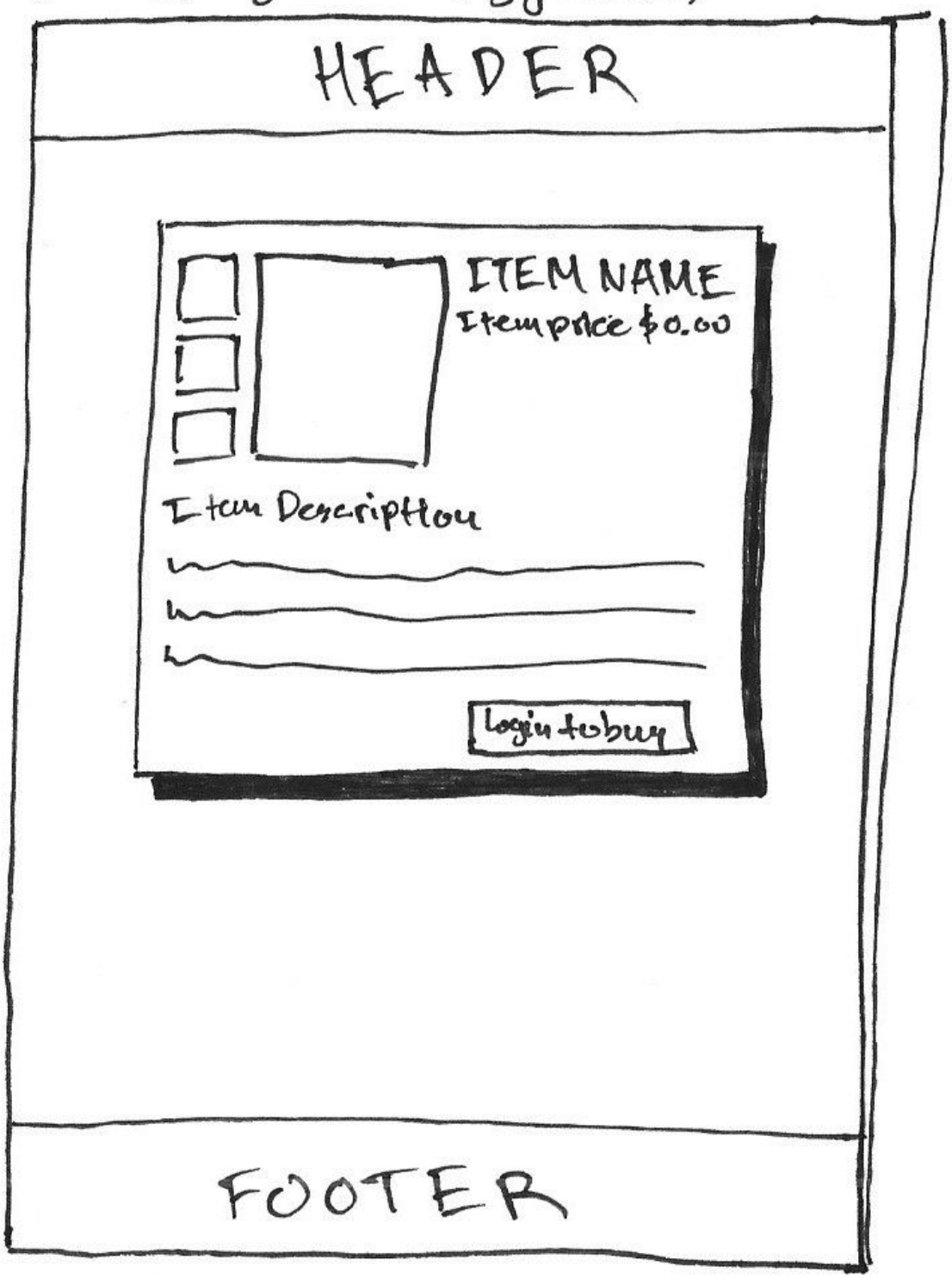
4. NON FUNCTIONAL SPECS V2

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 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 served from the team's account
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 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).
18. Gator Trader is supported in the following browsers: Chrome, Mozilla, Safari, and Internet Explorer.

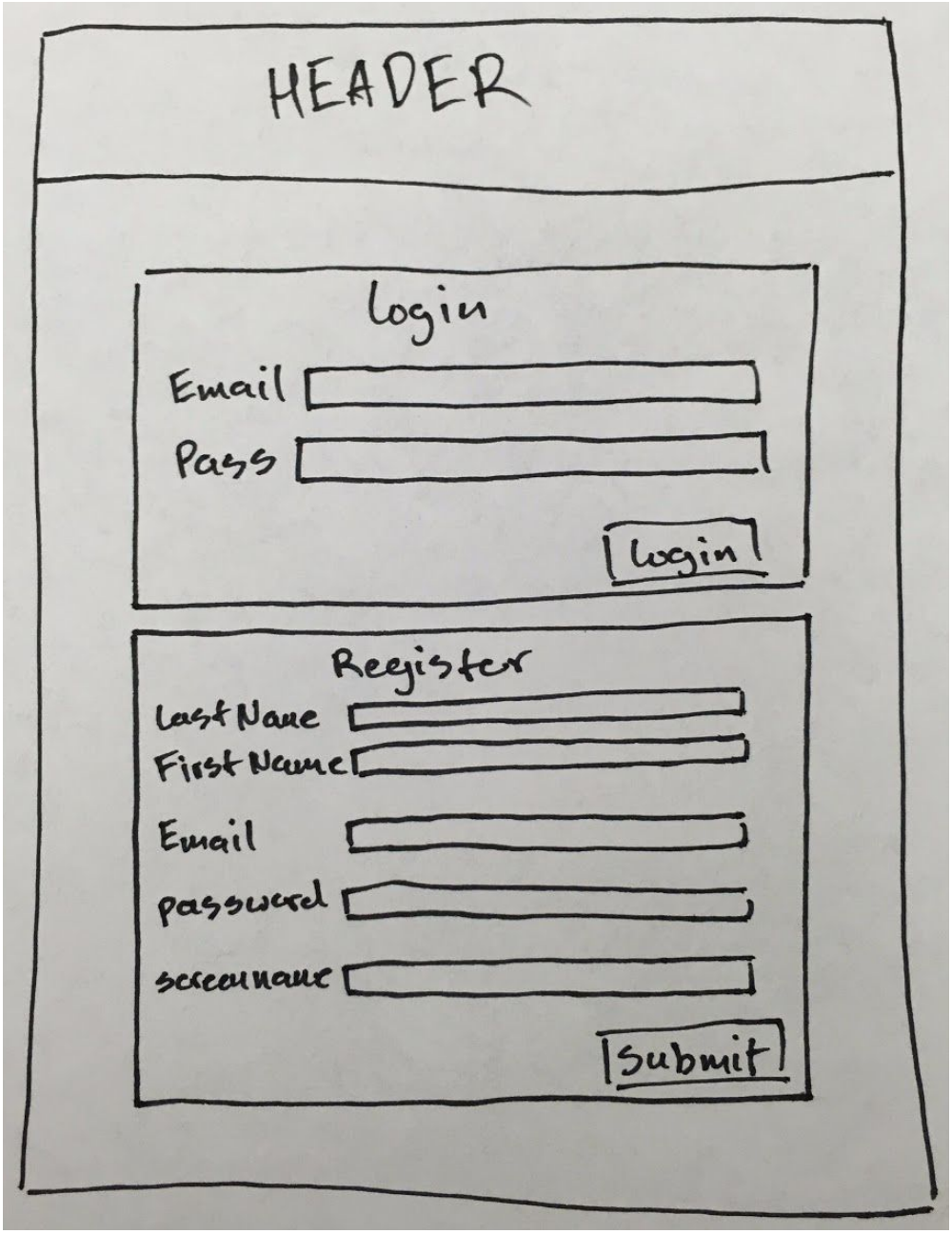
5. UI MOCK-UPS AND STORYBOARD



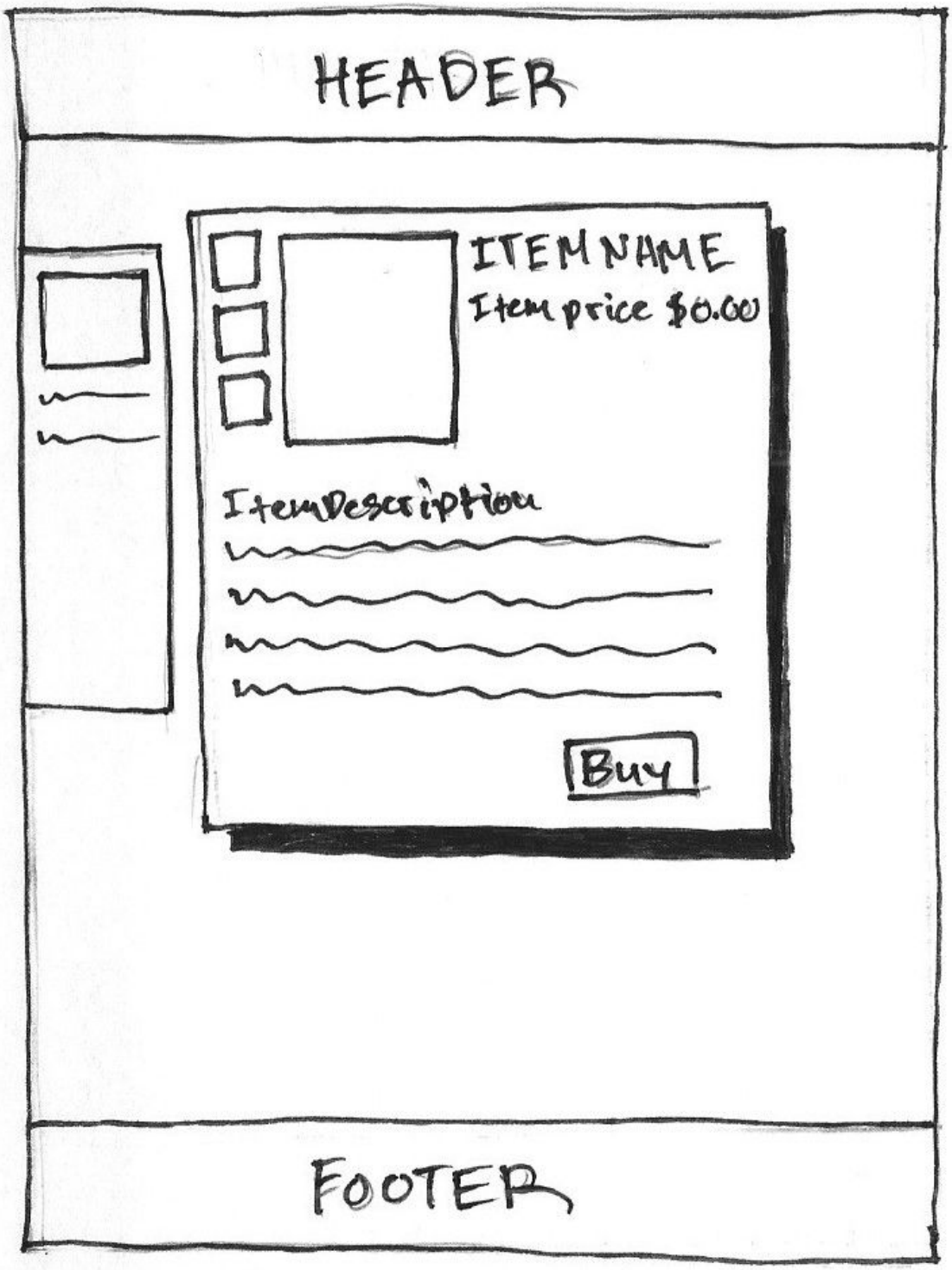
Item page (Not logged in)

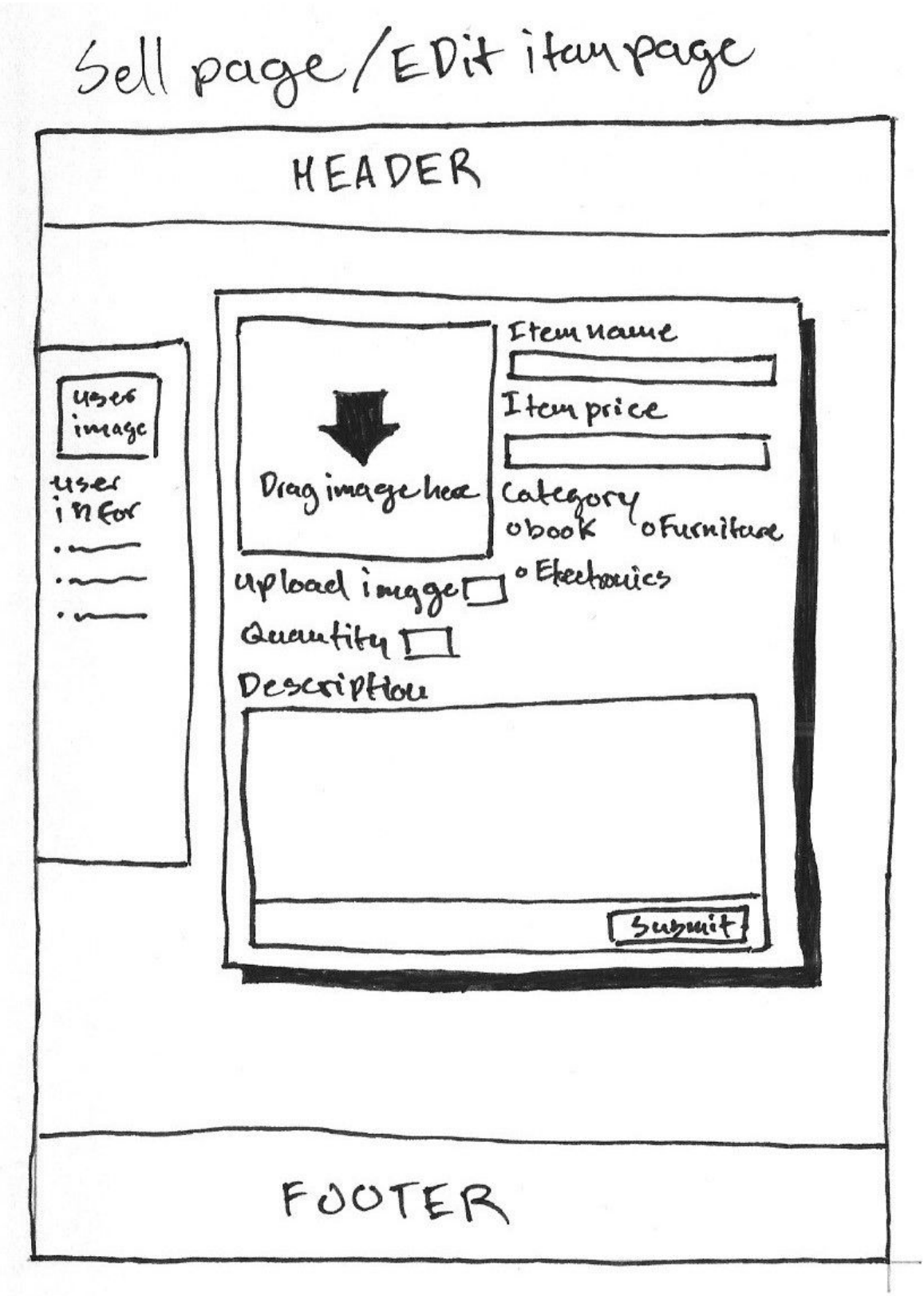


Login Page

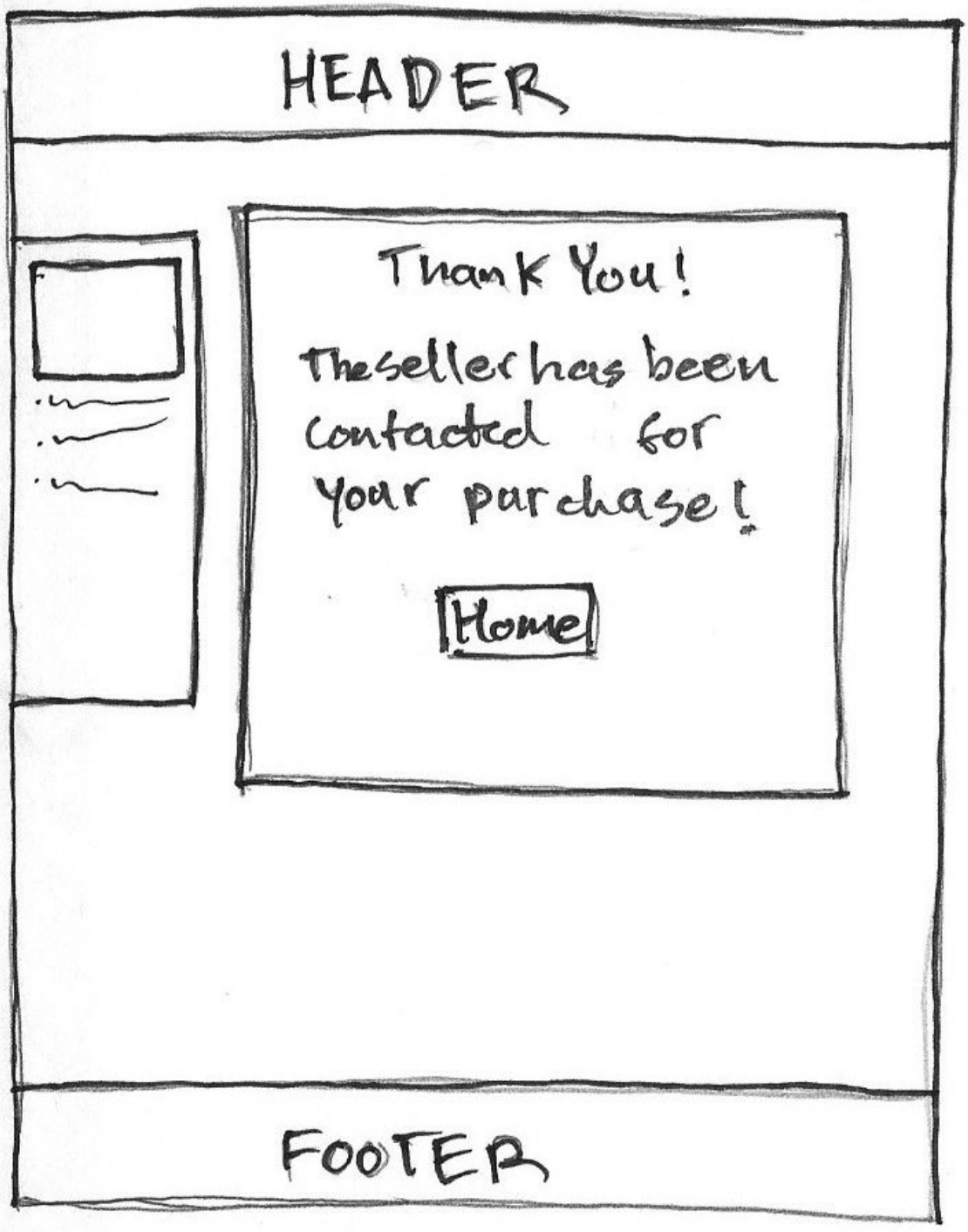

HEADER
login
Email
Pass
login
Register
Last Name
First Name
Email
password
screenname
Submit

Item page (logged in)

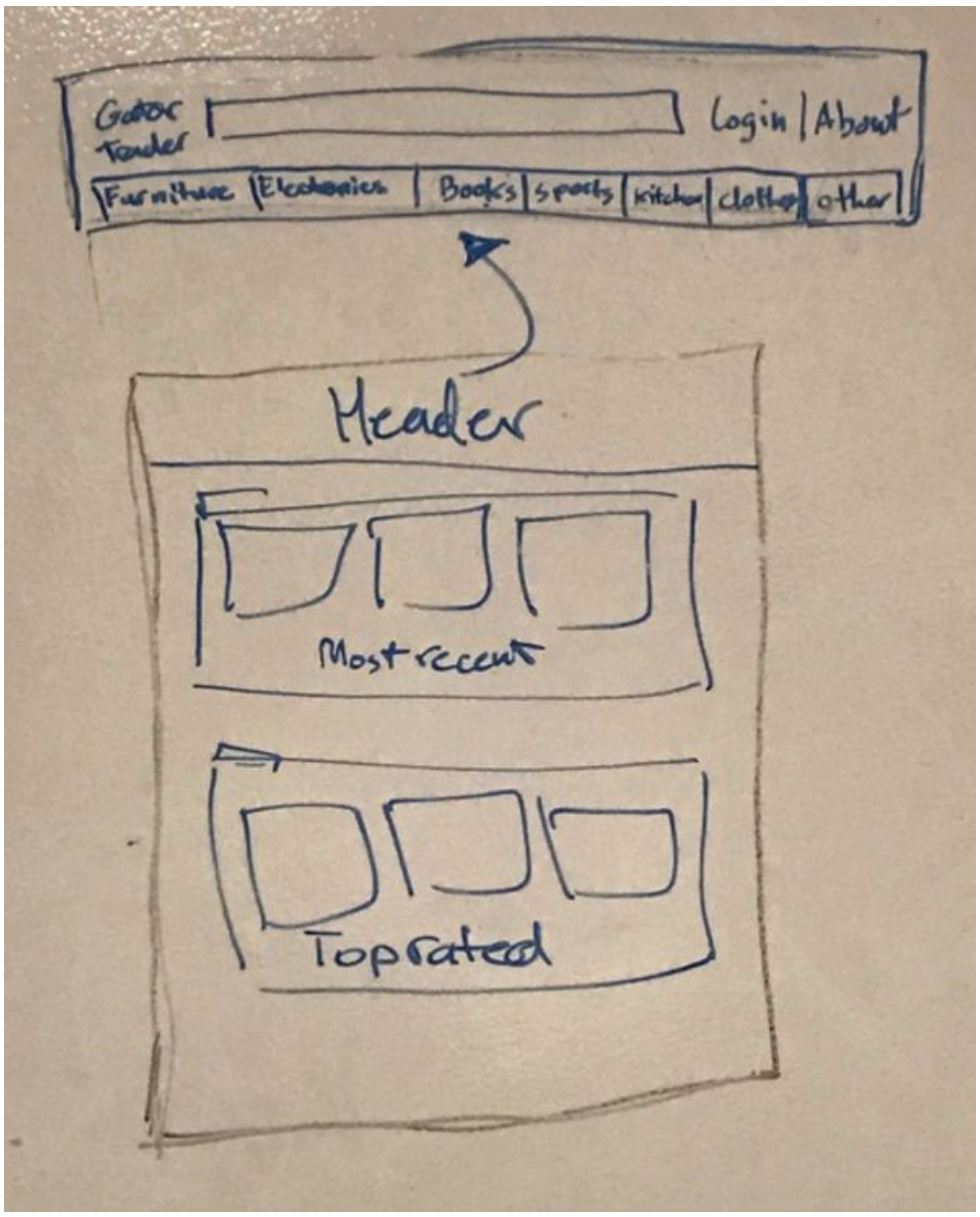



Sell page/EDit item page
HEADER
user image
user infor
Drag image here
Item name
Item price
Category
obook
oFurniture
oElectronics
upload image
Quantity
Description
Submit
FOOTER

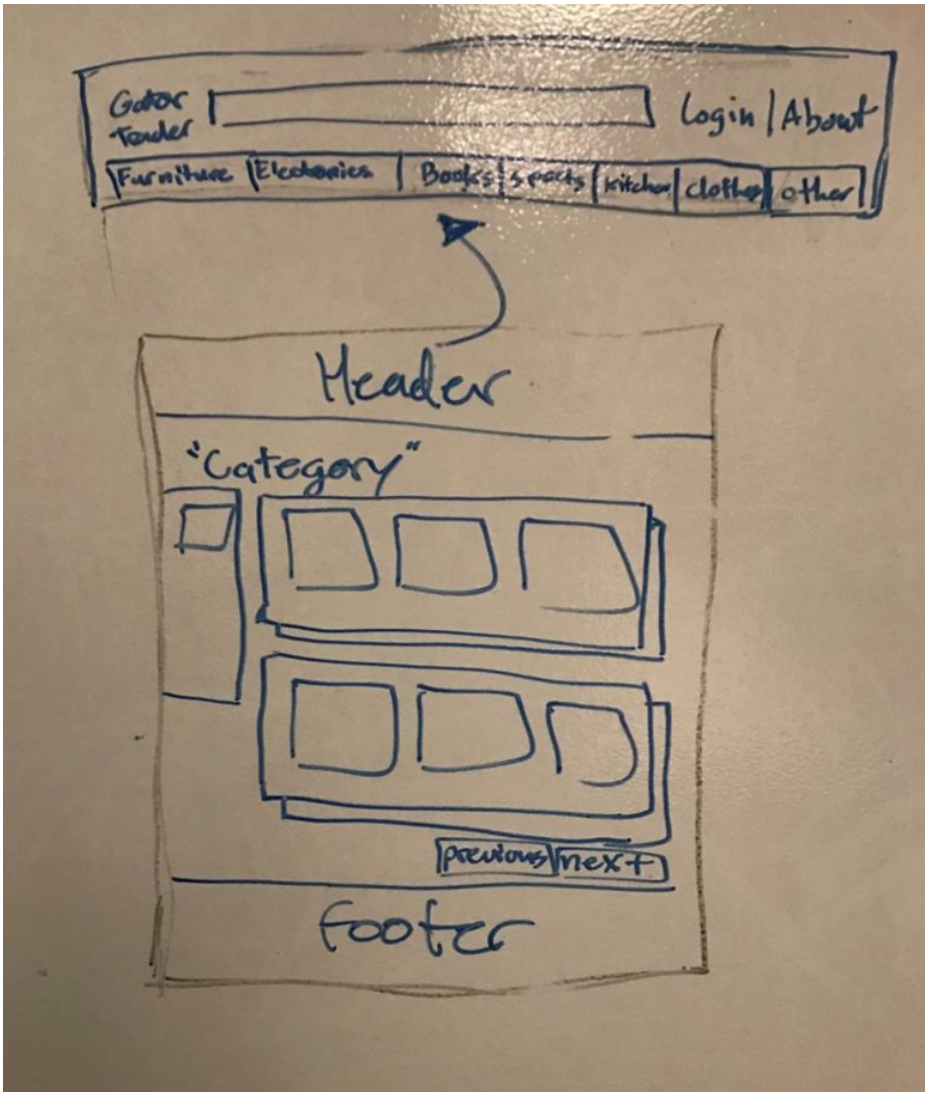
Purchase Notification



Homepage and Header (Not Logged In)



“Category” Page



6. HIGH LEVEL ARCHITECTURE

PHP FRAMEWORK

- CakePHP

VERSION CONTROL

- Git
- GitHub

WEB SERVICE

- Linux
- Apache
- MySql with Workbench
- PHP

WEB DEVELOPMENT

- HTML/CSS
- Foundation
- jQuery ?
- JavaScript ?

APIs

- Google Maps
- Google Analytics
- Google ADMIN Reports ?
- Google CustomSearch ?

IDE

- NetBeans

BROWSERS SUPPORTED

- Google Chrome
- Safari
- Internet Explorer
- Mozilla Firefox

SERVER

- Amazon Cloud

DATABASE ORGANIZATION

USERS
id: int(11) AI PK
last_name: varchar(45)
first_name: varchar(45)
email: varchar(255)
password: varchar(255)
screen_name: varchar(45)
is_admin: tinyint(4)
is_seller: tinyint(4)

ITEMS
id: int(11) AI PK
user_id: int(11) Foreign key from Users>id
title: varchar(100)
category: varchar(45)
description: varchar(1000)
price: decimal(4,2)
img1: tinyint(4)
img2: tinyint(4)
img3: tinyint(4)
img4: tinyint(4)
date_posted: timestamp

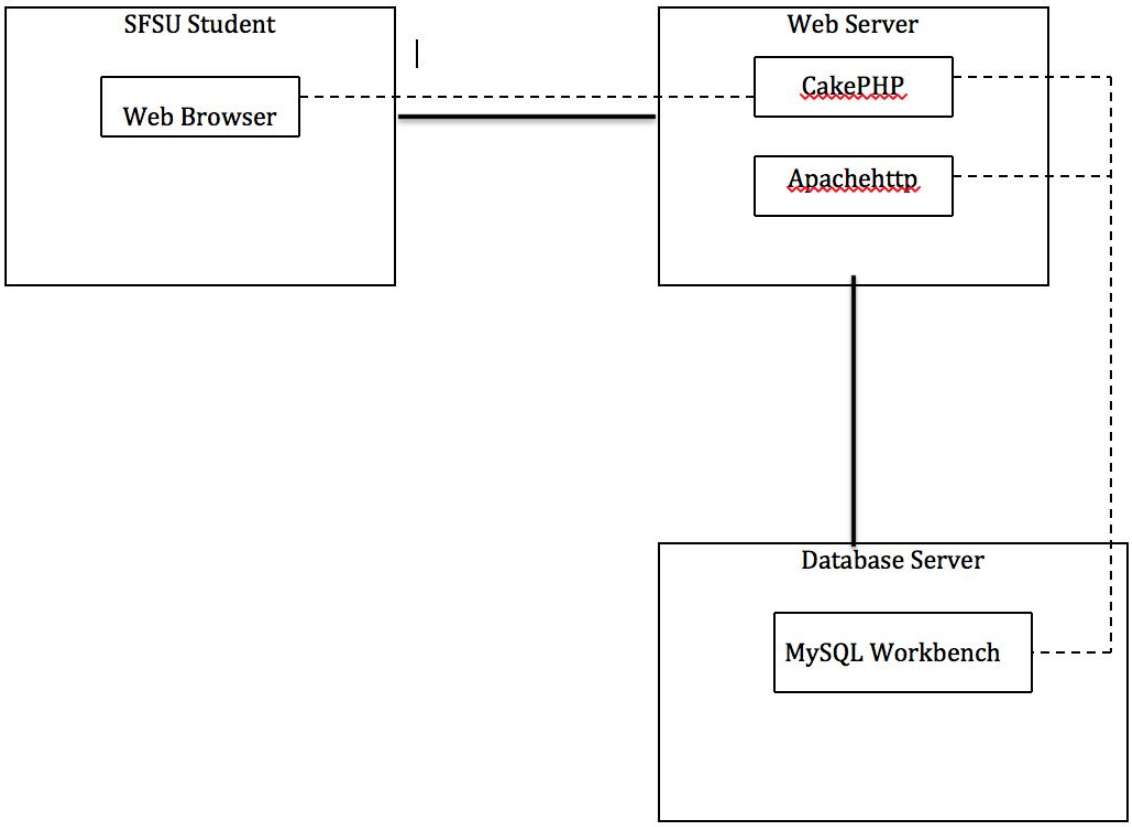
Algorithms

Search Algorithm: Only the items listed in the database shall be searched. Searching will be done by a percent match of the search item against the concatenation of each line of the item database.

Sorting Algorithm: Sorts the results of the searched keywords by price, most recent, etc., either from low to high or high to low, depending on the user's choice.

7. HIGH LEVEL UML DIAGRAMS

UML Component and Deployment Diagram



8. KEY RISKS

Teamwork Risks

- Communication
- Can’t always meet at the same time frames

Technical Risks

- Unable to connect to workbench
- CakePhp is difficult to work with

Schedule

- Vertical Prototype
- Meeting only once a week
- Delivering product on time

Resolving the Risks

- Figuring out more times to meet up
- Keep communication constant through Slack
- Have the teammates who have been learning CakePhp teach the others who don’t know how to use it
- Have the teammates who got workbench to work teach it to the others who couldn’t
- Get together to figure out how to do the vertical prototype
- Make sure to meet up before the deadline to make sure it gets done

9. TEAM ORGANIZATION

Rebecca Stankus
Team Lead and Database ADMIN

Farbod
CTO and Backend Developer

Kai
Documentation and Backend Developer

Kyle
Chief Spokesperson and Backend Developer

Krunal
Documentation and UI Developer

Jeremy
Git/GitHub Admin and UI Developer