Final Project SW Engineering CSC 648-848 Summer 2016
Gator Market

Group #2 (The Fighting Mongooses)
Jeffrey Noehren - jeffreynoehren9@gmail.com
Jason Newman
Olatope Oladipupo
Kenny Tran
Maria Lienkaemper

http://sfsuswe.com/~su16g02/CSC648Group02/mini/

08/09/16

SW Engineering CSC 648/848 Summer 2016

Gator Market

The Fighting Mongooses (Group 2)

Jeffrey Noehren jeffreynoehren9@gmail.com
Jason Newman gojirra@gmail.com
Olatope Oladipupo
Kenny Tran
Maria Lienkaemper

Milestone 1

June 28, 2016

Executive Summary

Gator Market allows students to sell their unwanted items on a simple to use website. After the end of a semester, many students have books from their classes that they no longer need or they're moving out of their housing and need to sell extra furnishings. Gator Market allows for these students to quickly sell their items to student who need those books for the next semester or have just moved to and need to furnish their living space. Gator Market allows users to easily browse and search for items that other student users have posted. Gator Market blends the familiarity of the commerce websites that we are all used to with the novelty of being exclusively for San Francisco State students. This narrow market saves both buyers and sellers time by allowing direct contact to fellow students with the same product needs. The users will be able to search and filter results by attributes, such as price, to find the item that they want to purchase. Gator Market won't nag users with login screens; users will be able to browse the available items before signing up.

Use Cases

Use Case 1 - **Buyer**

Claudia is a first year college student at SFSU. She is looking for a place where she can easily search for textbooks for her required classes other than the campus bookstore. She discovers the On-line buy/sell website for SFSU students through her classmates. Upon entering the website, the landing page is the first thing she sees and is immediately shown pictures of textbooks, furniture, and other items as well the newest and latest selling posts made by other students. She searches for her textbook in the search bar and the search results are displayed on a page with the title of the textbook, price, and user. She sorts the results by the lowest price and clicks the cheapest one that is displayed at the top of the list. Here she checks the uploaded image of the textbook to verify the condition. After confirming all the details, she clicks the buy button.

Use Case 2 - **Seller**

Joe is a graduating undergraduate Computer Science major at SFSU. Joe is looking into selling his textbooks as he no longer needs them. He is aware that there is an On-line buy/sell website for SFSU students because he lived through the process of creating it the semester before. Upon entering the website, the landing page with images of textbooks and categories displaying the newest and latest list postings by other students is the first thing greeted to the user. Joe browses the page and finds the button to make a selling post. He clicks the post button and is brought to a page where he must include an image of his textbook, price, description, and set the tag to category of book to be related to Computer Science, condition of textbook, and delivery option. After filling out all the necessary required steps, Joe is able to post his listing to the public on the website where it will show a timestamp date of when he posted the listing as well as his username and price.

Use Case 3 - **Unregistered User**

Joan is a student in her last year of high school doing a college visit on the campus. Joan came across the On-line buy/sell website for SFSU students through one of the college campus advisors. She enters the URL and is brought to the landing page of the website where she can see images of textbooks as well as the newest and latest postings made by the other students attending SFSU. She browses through one of the categories and clicks on one the postings of a bed being sold. She attempts to click on the username of the seller and is immediately redirected to a login page where she has to enter in her information in order to view the information before she can do anything else.

Use Case 4 - **Website Administrator**
Daniel is a Website Administrator for the On-line buy/sell website for SFSU students. He maintains and keeps uphold of the rules and regulations of the website. While he is scrolling through the newest listings of items for sale on the landing page, he notices a list posting with inappropriate content in it. As the Website Administrator, Daniel proceeds to open up the posting and completely remove it from website with the options that are only viewable with administrative privileges. The user is notified immediately with an automated message sent to their e-mail regarding the reason of their list post being removed.

Data Definition

Gator Market: The name of our website.
Users: Users are people who visit the site. Users consist of Registered Users and Non-registered Users.
Create an Account Button: A button which allows a User to create a User Account.
Registered User: A user with a User Account.
Non-registered User: A user which does not have an account, but is browsing the site.
User Account: The information stored about a Registered User. A User Account is required for a user to post listings or buy items. User Accounts shall consist of the following data:

- User Name
- Email Address
- Type (Standard or Admin)
- Rating
- Password
- Current listings

Standard User: A user without administrative privileges.
Admin User / Admin: A member of the site with administrative privileges such as removing listings and banning users.
Seller: The user that posted a specific listing (See Listings below).
Seller Rating: Indicates the overall satisfaction Buyers have had in dealing with this seller. Computed by averaging all ratings given to the Seller by Buyers who interacted with the Seller.
Buyer: The user that has clicked the "buy" button of a specific listing.
Item: The item being sold by a Seller, detailed in a Listing.
"Sell Something" Button: The button that is used to post a listing.
Listings: Listings are posted to the site by Sellers which contain the information about the item being sold. Listings contain the following data:

- Image(s)

- Price
- Description
- Seller
    - Seller Rating
- Tags (Books, Furniture, Misc)
- Timestamp
- Condition (Poor, Good, Like New, New, etc).
- Delivery Options
    - Willing to Mail
    - Campus Meetup
    - Pick-up

Buy Button: The button within a Listing which is used to initiate the buying process.
Buying Process: The seller will be notified that there is a buyer for their listed item, and they may begin communication through in-site messages or email (depending on what we are able to implement) to determine the specifics of the transaction such as Delivery Options. More specifics TBD at a later date.
Willing to Mail: The Seller of an Item is willing to ship the item to the Buyer.
Campus Meetup: The Seller of an Item is willing to deliver the item in person.
Pick-up: The Seller is willing to allow the Buyer to come in person to pick up the item from a certain address given by the Seller.
Landing Page: The front page of the site.
Browsing: When a User browses the site for listings. May be used to refer to a non-member that is using the site to view listings but is not buying or selling.
Ban: When an Admin deletes the account of a User for violating the Gator Market EULA.
Post: When a User creates a new Listing.
Delete: When a user removes their own or an Admin removes a listing from the site.

Functional Specs

1. The landing page shall be available for browsing
    1.1 Browsing options to sort by shall be:
        1.1.1 Newest
        1.1.2 Most-Least Expensive
        1.1.3 Least-Most Expensive
        1.1.4 Most Popular
2. There shall be a search bar on each page of the website to search for specific items
3. Visitors to the site shall be able to browse listings without creating an account.
    3.1 Listings shall include:
        3.1.1 Username of the seller.
        3.1.2 Seller's rating
        3.1.3 Price
        3.1.4 Image(s)
        3.1.5 Posting Date
        3.1.6 Description
        3.1.7 Delivery Options

3.1.7.1 Pickup/Dropoff

3.1.7.2 Campus meet up

3.1.7.3 Mail

3.1.8 Descriptive Tags

3.1.8.1 Books, furniture, appliances

3.1.9 Condition

4. User shall be able to create an account

4.1 Shall be able to post a listing

4.1.1 Shall be able to upload an original picture and describe item.

4.1.2 Shall be able to delete their own listings.

4.2 Shall be able to report other users (a message shall be sent to admins).

5. Admin shall be able to create an account

5.1 Shall be able to delete listing if deemed in appropriate

5.1.1 If listing is deleted the user shall be contacted with the reason why

5.2 Shall be able to delete user accounts due to inappropriate listings or other violations of our EULA.

6. System shall automatically delete listings that are a month old or older.

7. There shall be a user rating taken from other users who have had an interaction with

Non-functional Specs

1. Application shall be developed using pre-approved set of SW development and collaborative tools provided in the class. Any other tools or frameworks have to be explicitly approved by Marc Sosnick on a case by case basis.

1.1 Application shall be developed using class provided LAMP stack

1.2 Application shall be hosted and deployed on Amazon Web Services as specified in the class

1.3 Data shall be stored in the database on the class server in the team's account

1.4 Application shall be served from the team's account

1. 5 Google analytics shall be added for major site functions.

2. Application shall be optimized for a standard desktop/laptop browsers, and shall render correctly on the two latest versions of all major browsers: Mozilla, Safari, Chrome and IE. It shall degrade nicely for different sized windows using class approved programming technology and frameworks

3. No more than 50 concurrent users shall be accessing the application at any time

4. Site security: Basic best practices to be applied (as covered in the class)

4.1 Privacy of users shall be protected and all privacy policies will be appropriately communicated to the users.

5. The language used shall be English.

Application shall be very easy to use and intuitive. No prior training shall be required to use the website.

6. Messaging between users shall be done only by class approved methods to avoid issues of security with e-mail services.

7. Pay functionality (how to pay for good): shall be simulated this with proper UI, no backed.

8. Modern SE processes and practices must be used as specified in the class, including collaborative and continuous SW development, using the tools approved by instructors

9. The website shall prominently display the following text on all pages "SFSU/FAU/Fulda Software Engineering Project, Summer 2016. For Demonstration Only". (Important so as to not confuse this with a real application).

Competitive Analysis

	Craigslist	OfferUpNow	Hobbly	Gator Market
For SFSU Student Exclusively	X	X	X	✓
Multiple Delivery Options	X	X	X	✓
Seller Rating	X	X	X	✓
Search Categories	✓	✓	✓	✓
Ease and Simplicity	✓	✓	X	✓

Key:
✓: Feature is included
X: Feature is not included

Through the competitive analysis, we found that there are a lot of websites out there for people to post things to sell and anybody can browse and buy. The problem with a lot of these websites is it is hard to tell rather or not the sellers are trusted, and some of them are not very easy to use. The plan for our website is to create a site exclusively for SFSU students where the sellers can be trusted, through a seller rating feature. Also the competitor websites do not specify how the buyer and seller would exchange the item. We hope to give the seller different options for how they would like the items to be exchanged so the user is easily able to tell in advance the delivery options, rather it will be mailed, picked up, dropped off, or there will be a campus meet up.

High-Level System Architecture

- Linux - Operating System that provides a base foundation on our server instance for all the other systems to run. We are running Ubuntu 16.04

- Apache - Web server software that handles user interaction and fetches requests from the server. For example when the user goes to the home page
- MySQL - Database to hold information. We will be using our database to store login information about the user as well as the products we will be listing
- PHP - Used for generating dynamic web pages. When the user makes a request that requires PHP, Apache sends the request to PHP and waits for it to generate an HTML file to send back to the user. For example when the user logs onto our webpage we will need to make PHP calls to the database to check to see if the user is there and generate the correct page based on their action.
- Model-View-Controller - We will be using the MVC framework which is divided into three parts, Model, View and Controller. The View is what is actually displayed on your screen when you view a website, and it changes based on the Model. The Model collects and stores information that the Controller instructs. And the Controller gets information from the user and updates the the Model.

Team

Jeffrey Noehren - Team Lead/Backend
Jason Newman - Tech Lead/Frontend
Kenny Tran - Backend
Olatope Oladipupo - Frontend
Maria Lienkaemper- Backend

SW Engineering CSC 648/848 Summer 2016

Website:

The Gator Market

Group #2 - The Fighting Mongooses:

Jeffrey Noehren jeffreynoehren9@gmail.com

Jason Newman

Olatope Oladipupo

Kenny Tran

Maria Lienkaemper

Milestone/Date:

Milestone 2

July 15, 2016 Version 1.0

Table of Contents

1. Executive Summary

Gator Market allows San Francisco State University students to sell their unwanted items on a simple to use website. After the end of a semester, many students have books from their classes that they no longer need or they're moving out of their housing and need to sell extra furnishings. Gator Market allows for these students to quickly sell their items to student who need those books for the next semester or have just moved to and need to furnish their living space. Gator Market allows users to easily browse and search for items that other student users have posted. Gator Market blends the familiarity of the commerce websites that we are all used to with the novelty of being exclusively for San Francisco State students. This narrow market saves both buyers and sellers time by allowing direct contact to fellow students with the same product needs. The users will be able to search and filter results by attributes, such as price, to find the item that they want to purchase. Gator Market won't nag users with login screens; users will be able to browse the available items before signing up.

2. Use Cases

Use Case 1 - Buyer

Claudia is a first year college student at SFSU. She is looking for a place where she can easily search for textbooks for her required classes other than the campus bookstore and other items that she will need to live comfortably in a new city such as a couch and a bike. She discovers **The Gator Market**, an on-line buy/sell website only for SFSU students through her classmates. Upon entering the website, the **landing page** is the first thing she sees. She searches for a couch in the search bar and is taken to a page with **listings** that she can **browse** through. She is able to sort them by lowest price by using the **categories/tags**. She clicks a **post** and checks the uploaded image of the couch to verify the condition. After confirming all the details, she clicks the buy button where she is prompted to make an account and taken to the **account creation page** to create an account. She then can pick between the options of either a **campus meetup** or **willing to mail** which puts her through the **buying process** between her and the seller.

Use Case 2 - Seller

Joe is a graduating undergraduate Computer Science major at SFSU. Joe is looking into selling his textbooks because he no longer needs them. He is aware of the **The Gator Market** website and its purpose. Upon entering the website, the **landing page** with images of textbooks and **categories** displaying the newest and latest list **postings** by other students is the first thing greeted to the **user**. He is a **registered user** so he logs into the website. Joe then goes to **post** a **listing**. He is brought to the **listing creation page** where he'll be able to enter information about his **listing**. After filling out all the necessary required steps, Joe's **listing** is now viewable by the public on the website where it will show a timestamp date of when he posted the **listing** as well as his username and price. Joe will then be able to edit and **delete** his **listing** when logged in.

Use Case 3 - Unregistered User

Joan is a student in her last year of high school doing a college visit on the campus and is an **unregistered user**. Joan came across **The Gator Market** through one of the college campus advisors. She enters the URL and is brought to the **landing page** of the website where she can **browse** the newest and latest **listings** made by the other students attending SFSU. She **browses** through one of the **categories** and clicks on a **listing**. She clicks on the buy button and is immediately redirected to the **account creation page** where she needs a valid SFSU ID in order to create an account to view the information before she can do anything else.

Use Case 4 - Administrator

Daniel is an **Administrator** for **The Gator Market**, an On-line buy/sell website for SFSU students. He maintains and keeps uphold of the rules and regulations of the website. While he is scrolling through the newest **listings** of items for sale on the **landing page**, he notices a list posting with inappropriate content in it. As the **Administrator**, Daniel proceeds to open up the posting and **deletes** it from website with the options that are only viewable with administrative privileges. The **user** is notified immediately with an automated message sent to their e-mail regarding the reason of their list post being removed.

3. Data Definition

The Gator Market: The name of our website.
Users: Users are people who visit the site. Users consist of Registered Users and Unregistered Users.
Unregistered User: Users that do not have any additional functions aside from being able to browse and search the site for listings.
Registered User: A user with a User Account (SFSU students only) that can post listings or buy items.
Account Creation Page: The page a user is taken to after clicking the "Create an Account" button. This is where a user shall enter their information to create a User Account. This page shall include the following fields: Desired User Name, Email Address, Password. The Desired User Name field shall notify the user if their desired user name is not available by searching the database, and shall not proceed with account creation until they specify a desired user name which is available.
User Account: User accounts are objects stored in the database. They contain the information stored about a Registered User. Only current students at SFSU may created a User Account and become a Registered User. A User Account is required for a user to post listings or buy items. User Accounts shall consist of the following data:

- User Name
- Student ID
- Email Address: Private information. Used for relevant communications with registered users.
- Type: Admin or non-Admin. See "Admin" below.
- Seller Rating
- Password
- Current listings

Administrator: A user of the site with administrative privileges such as removing listings and banning users. All other Registered Users do not have these privileges.
Seller: The user that posted a specific listing (See Listings below).
Seller Rating: Indicates the overall satisfaction Buyers have had in dealing with this seller. Stored in the User Account object in the database. Computed by averaging all ratings given to the Seller by Buyers who interacted with the Seller.
Buyer: The user that has clicked the "buy" button of a specific listing.
Listing: Listings are posted to the site by Sellers which contain the information about the item being sold. Listings will be objects stored in the database containing the following information:

- Title
- Image(s)
- Price
- Description

- Seller
    - Seller Rating
- Tags (Books, Furniture, Misc)
- Timestamp
- Condition (Poor, Good, Like New, New, etc).
- Delivery Options
    - Willing to Mail
    - Campus Meetup
    - Pick-up

Listing Creation Page: The page where a Registered User can create a new listing. The page shall include the following fields: Image(s), Price, Description, Tags, Condition, and Delivery Options.

Buying Process: The seller will be notified that there is a buyer for their listed item, and they may begin communication through in-site messages or email (depending on what we are able to implement) to determine the specifics of the transaction such as Delivery Options. More specifics TBD at a later date.

Willing to Mail: The Seller of an Item is willing to ship the item to the Buyer.

Campus Meetup: The Seller of an Item is willing to deliver the item in person.

Pick-up: The Seller is willing to allow the Buyer to come in person to pick up the item from a certain address given by the Seller.

Landing Page: The front page of the site. Contains a well organized list of categories; a clean and attractive visual display of Listings which can be sorted by "most recent", "most popular", etc.; a Create an Account Button; and various navigation links relating to our company such as "about", "contact", "EULA", etc.

Categories / Tags: Various labels that a Seller chooses when posting an item. Examples include: "Book", "Furniture", "Clothing", etc. These will be stored in the Listing object and used for searching and UI organizational purposes.

Browsing: When a User browses the site for listings. May be used to refer to an Unregistered User that is using the site to view listings but is not buying or selling.

Ban: When an Admin deletes the account of a User for violating the Gator Market EULA.

Post: When a User creates a new Listing.

Delete: When a user removes one of their own listings or an Admin removes a listing from the site.

4. Functional Requirements

Priority 1

1. Users shall be taken to the landing page upon entering the site.
2. Search bar shall be present on each page of the website for users to search listings.
3. Unregistered Users
    3.1 Unregistered Users shall only be able to browse and search for listings.
    3.2 Unregistered Users shall only be able to view the following:
        3.2.1 Name of listing for sale.
        3.2.2 Price
        3.2.3 Image of listing for sale.
    3.3 Unregistered Users shall be able to create a User Account with a valid SFSU ID.
4. Registered Users
    4.1 Registered Users shall be able to browse and search for listings.
    4.2 Registered Users shall be able to manage their User Account.
    4.3 Registered Users shall be able to post a listing that includes the following:
        4.3.1 Username
        4.3.2 Name of listing for sale
        4.3.3 Price
        4.3.4 Image(s)
        4.3.5 Posting Date
        4.4.6 Description
5. Administrators
    5.1 Administrators shall be able to manage all users registered to the site.
    5.2 Administrators shall be able to delete listings in violation of the EULA.
        5.2.1 Registered Users shall be notified with a reason as to why listing was deleted.

Priority 2

1. Unregistered Users
    1.1 Unregistered Users shall be able to view the following in listings.
        1.1.1 Posting Date
        1.1.2 Description
        1.1.3 Descriptive Tags
        1.1.4 Condition
2. Registered Users
    2.1 Registered Users shall be able to choose a delivery option for listing.
        2.1.1 Pick-up/Drop-off
        2.1.2 Campus meetup
        2.1.3 Mail
    2.2 Registered Users shall be able to categorised with descriptive tags.

2.2.1 Books
2.2.2 Furniture
2.2.3 Appliances
2.3.4 Miscellaneous

2.3 Registered Users shall be able to choose the condition of their listing.

2.3.1 Like New
2.3.2 Gently Used
2.3.3 Poor

2.4 Listings shall have a posting date.

2.5 Registered Users shall be able to delete their own listing.

3. Administrators

3.1 Administrators shall be able to delete User Accounts in violation of EULA.

3.2 System shall be able to automatically delete listings after a certain period of time.

Priority 3

1. Browsing options to sort by shall be:

1.1 Newest
1.2 Most-Least Expensive
1.3 Least-Most Expensive
1.4 Most Popular

2. There shall be a user rating taken from other users who have had an interaction with

5. Non-functional Requirements

1. Application shall be developed using pre-approved set of SW development and collaborative tools provided in the class. Any other tools or frameworks have to be explicitly approved by Marc Sosnick on a case by case basis.

    1.1 Application shall be developed using class provided LAMP stack

    1.2 Application shall be hosted and deployed on Amazon Web Services as specified in the class

    1.3 Data shall be stored in the database on the class server in the team's account

    1.4 Application shall be served from the team's account

    1. 5 Google analytics shall be added for major site functions.

2. Application shall be optimized for a standard desktop/laptop browsers, and shall render correctly on the two latest versions of all major browsers: Mozilla, Safari, Chrome and IE. It shall degrade nicely for different sized windows using class approved programming technology and frameworks

3. No more than 50 concurrent users shall be accessing the application at any time

4. Site security: Basic best practices to be applied (as covered in the class)

5. Privacy of users shall be protected and all privacy policies will be appropriately communicated to the users.

6. The language used shall be English.

7. Application shall be very easy to use and intuitive. No prior training shall be required to use the website.

8. Google analytics shall be added for major site functions.

9. Messaging between users shall be done only by class approved methods to avoid issues of security with e-mail services.

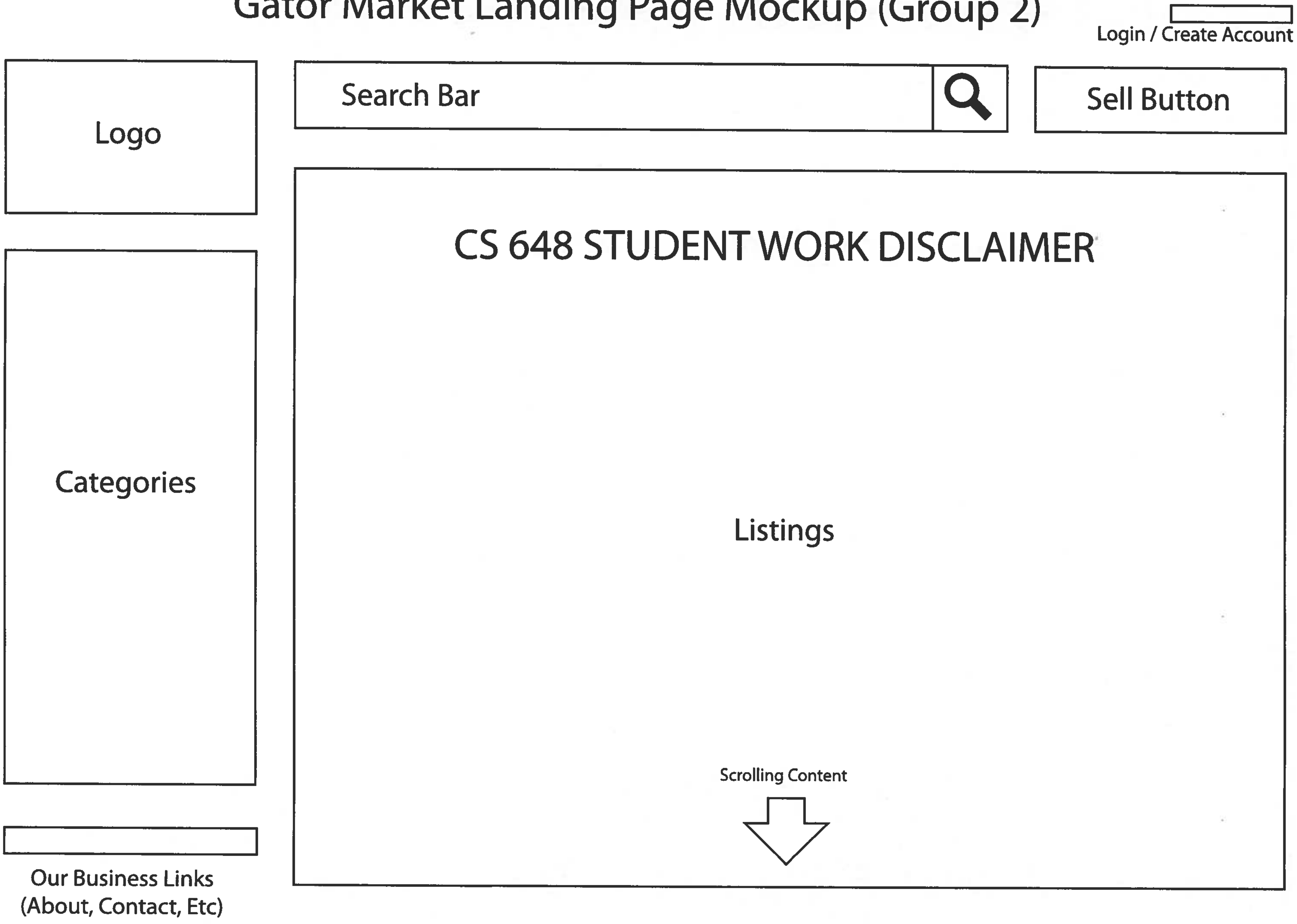
10. Pay functionality (how to pay for good): shall be simulated this with proper UI, no backed.

11. Modern SE processes and practices must be used as specified in the class, including collaborative and continuous SW development, using the tools approved by instructors

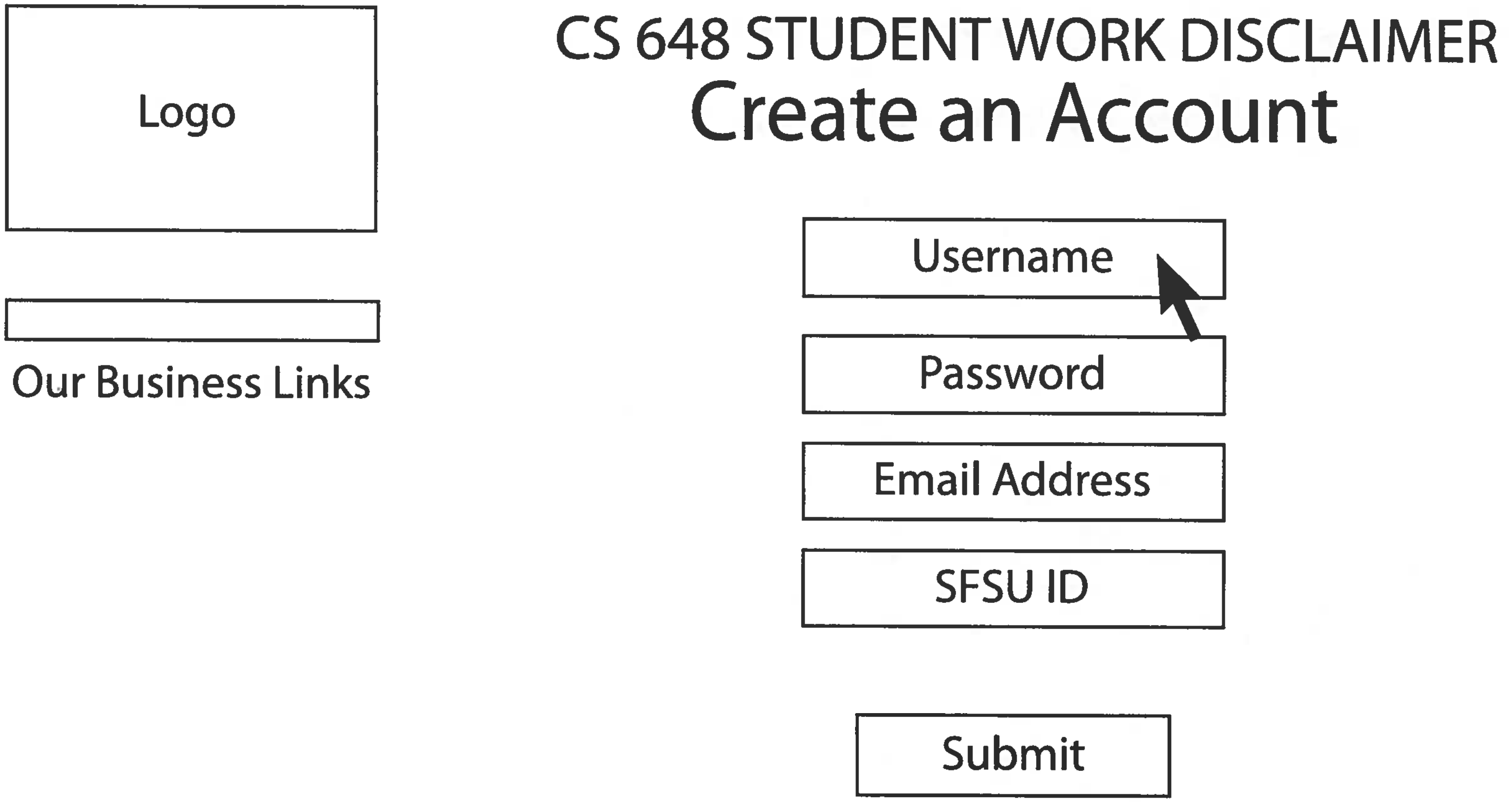
12. The website shall prominently display the following text on all pages "SFSU/FAU/Fulda Software Engineering Project, Summer 2016. For Demonstration Only". (Important so as to not confuse this with a real application).

6. UI Mock ups and StoryBoards

Gator Market Landing Page Mockup (Group 2)



Gator Market Create Account Page Mockup (Group 2)



Logo

Our Business Links

CS 648 STUDENT WORK DISCLAIMER

Create an Account

Username

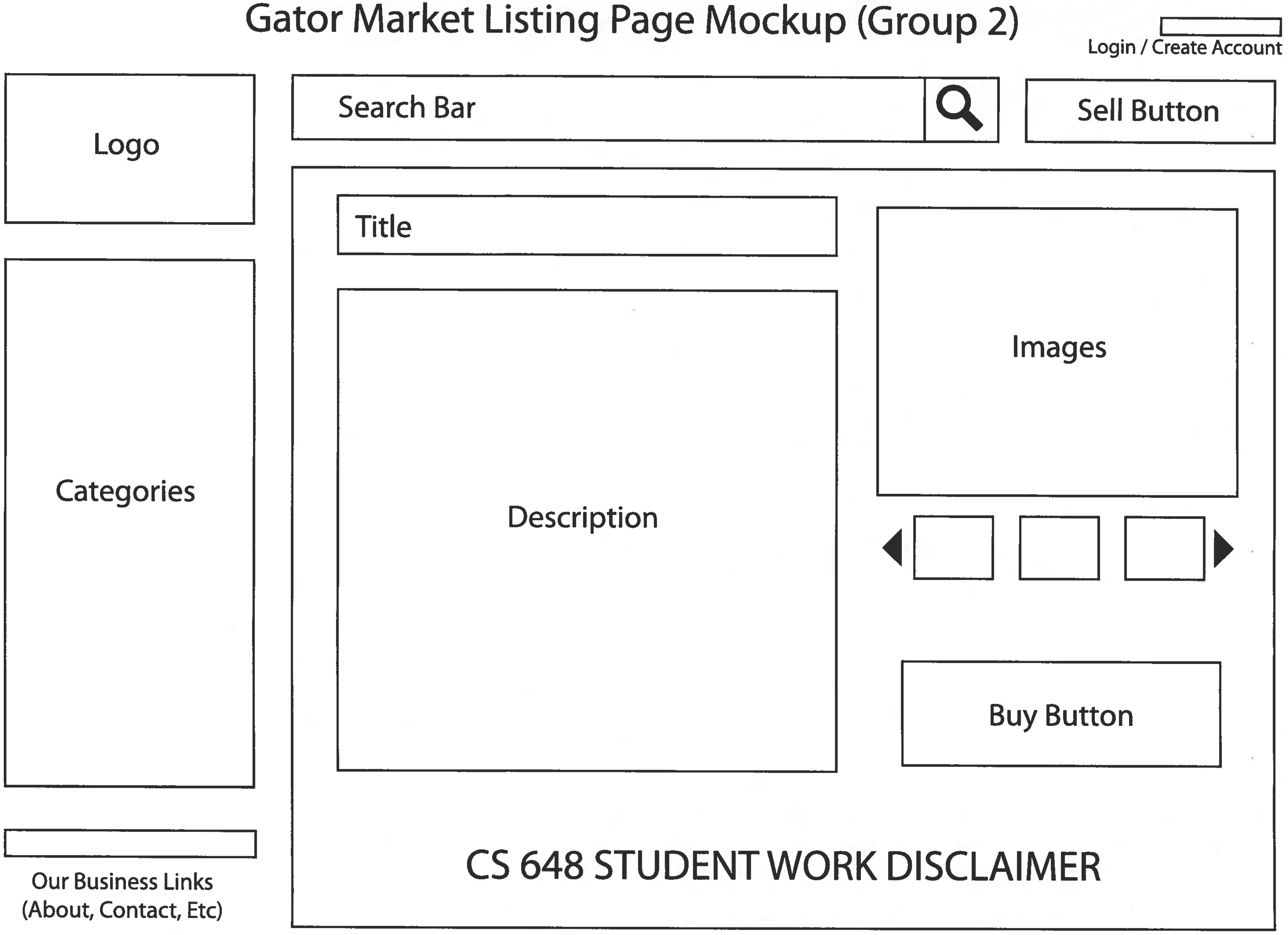
Password

Email Address

SFSU ID

Submit

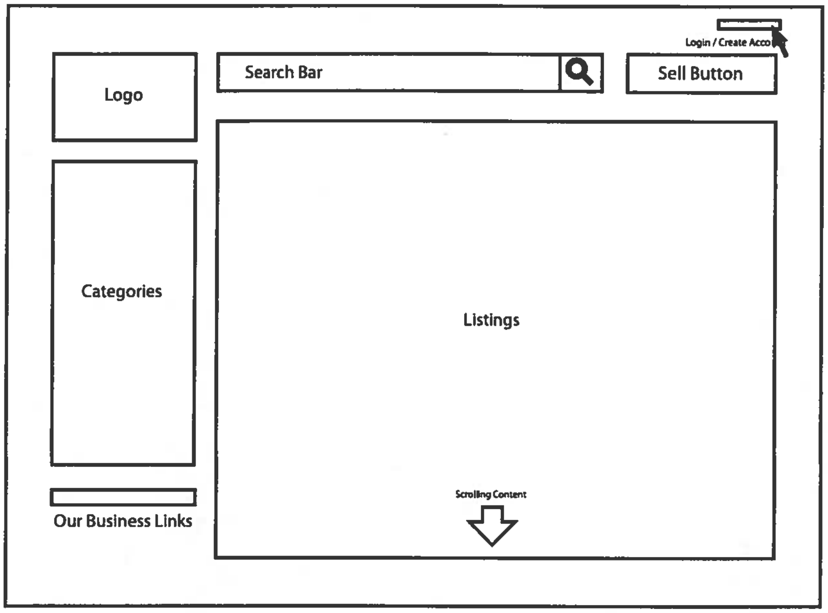
Gator Market Listing Page Mockup (Group 2)



Group 2

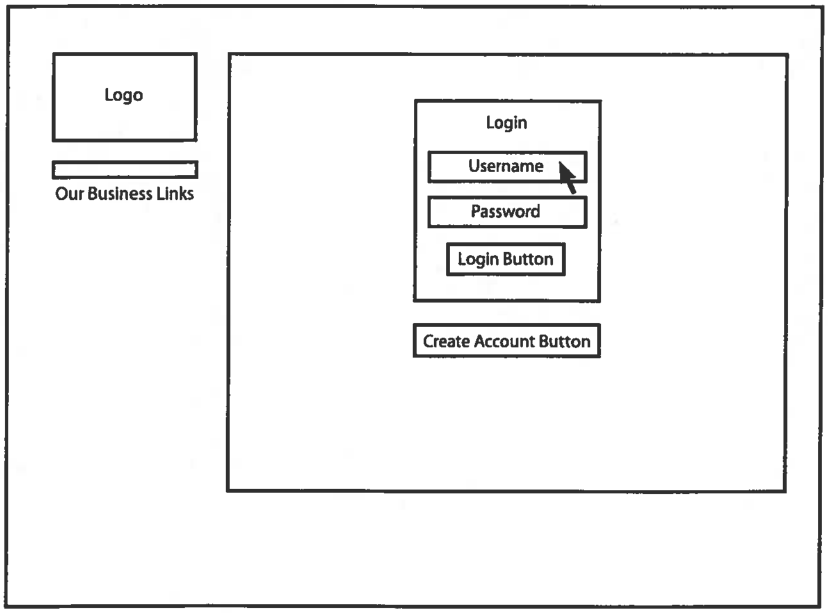
Gator Market Use Case: Admin

1) Landing Page



Admin arrives at the landing page, and then logs in.

2) Login Page



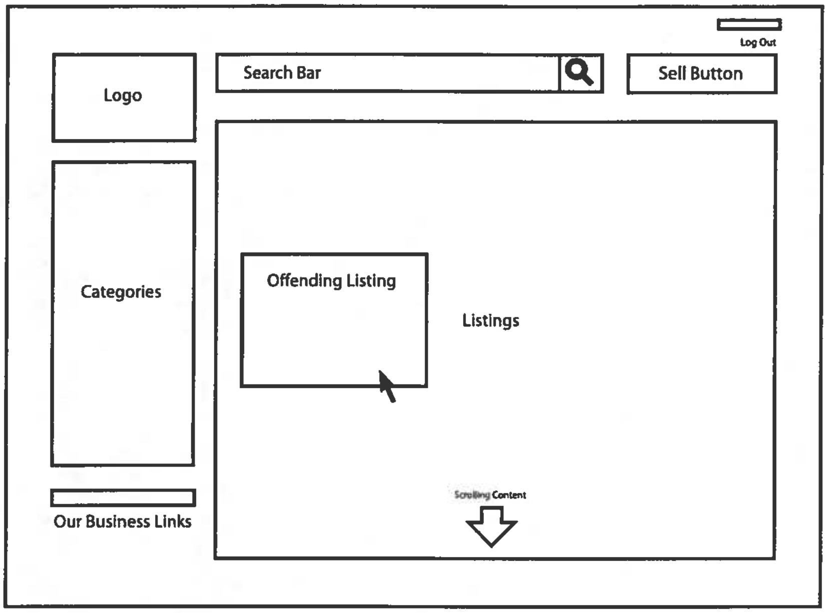
The Login Page allows a Registered User to log into their account, or an Unrestistered User to create a new account.

The admin fills out the fields Username and Password, and then clicks the login button, and then they are taken back to the landing page.

Group 2

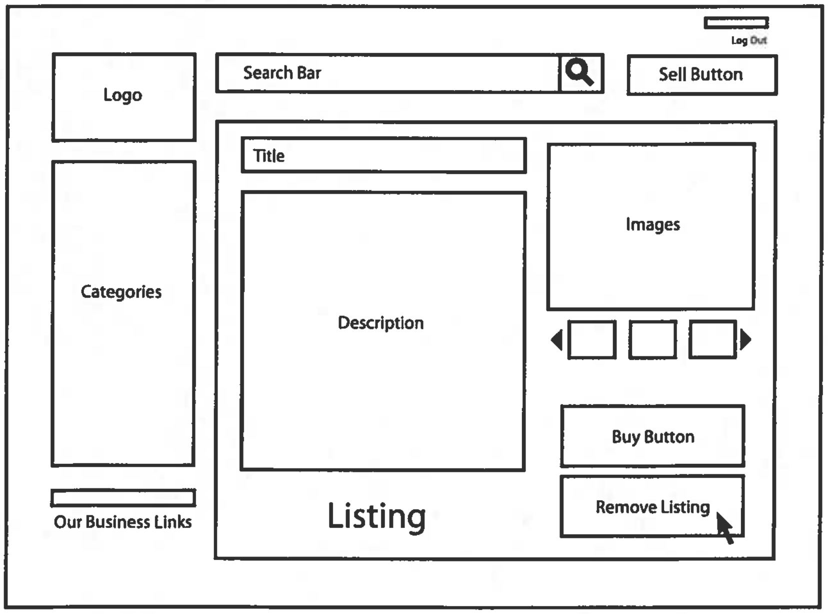
Gator Market Use Case: Admin

3) Landing Page



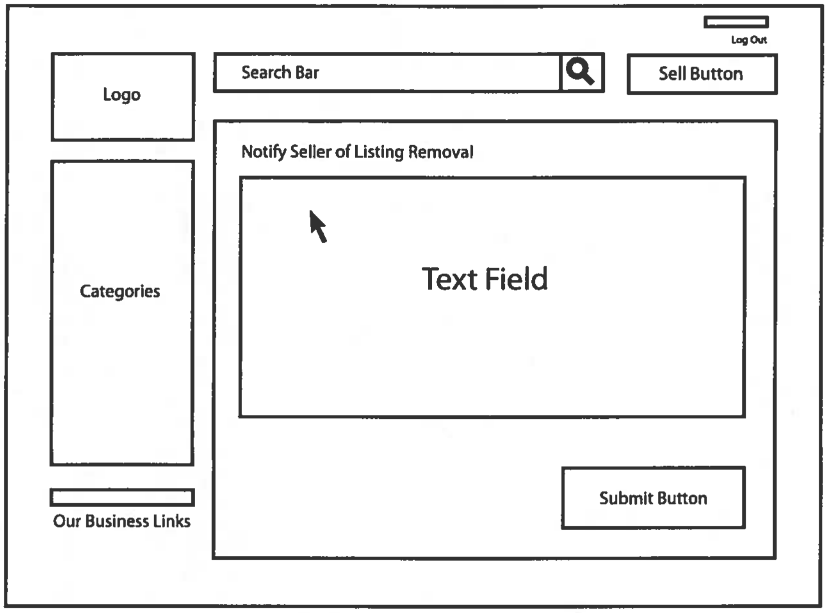
The admin sees a listing which breaks the rules of the site, and clicks it to be taken to the Listing page.

4) Listing Page



The admin clicks the Remove Listing Button and is then taken to the Notify Seller page.

5) Notify Seller Page

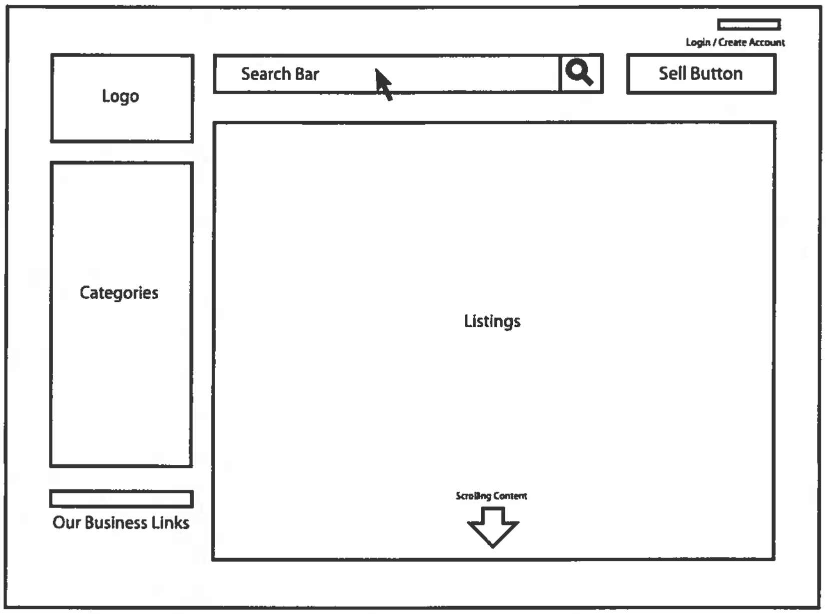


The admin fills out the text field with an explanation for why the Seller's Listing was removed and hits submit to send it to the Seller.

Group 2

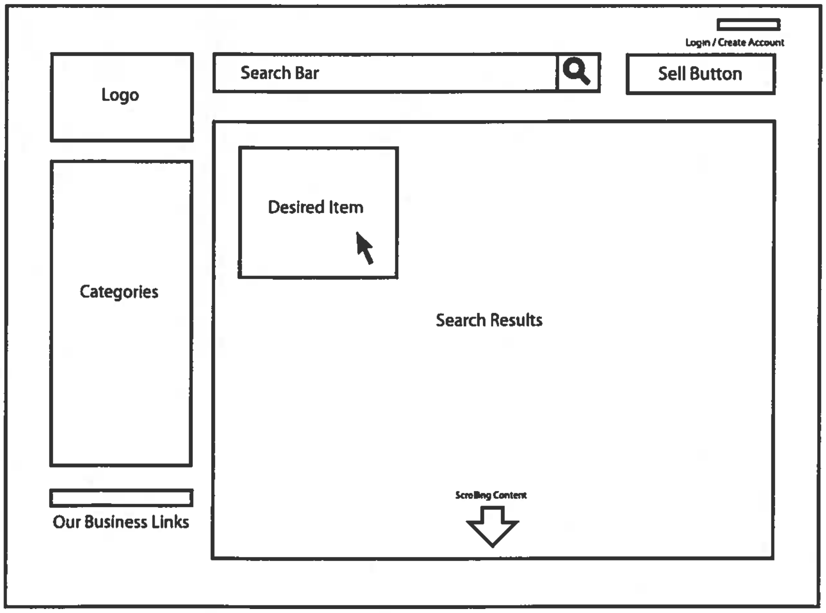
Gator Market Use Case: Buyer

1) Landing Page



User arrives at the landing page, and then searches for an item they are interested in buying using the Search Bar.

2) Search Results

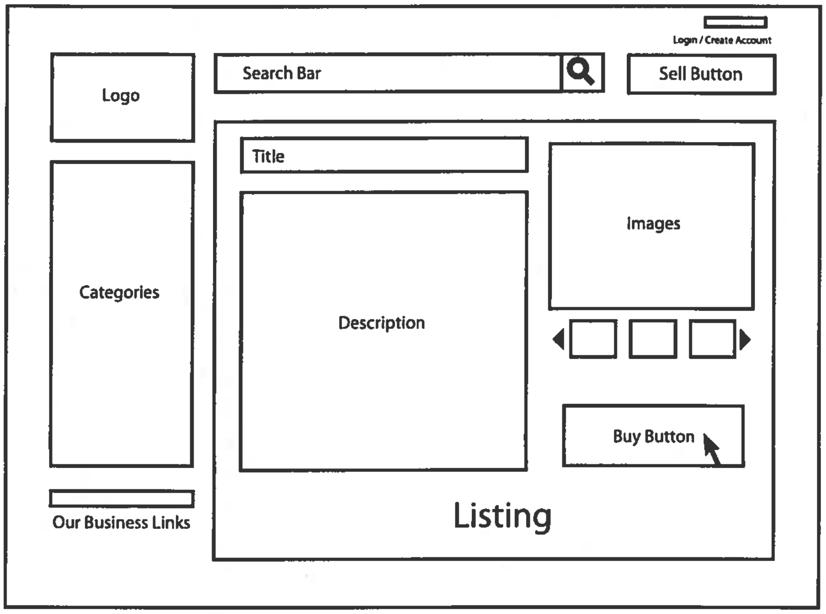


User is taken to a page that shows relevant Listings. The user then clicks on the desired Listing and is taken to the Listing page.

Group 2

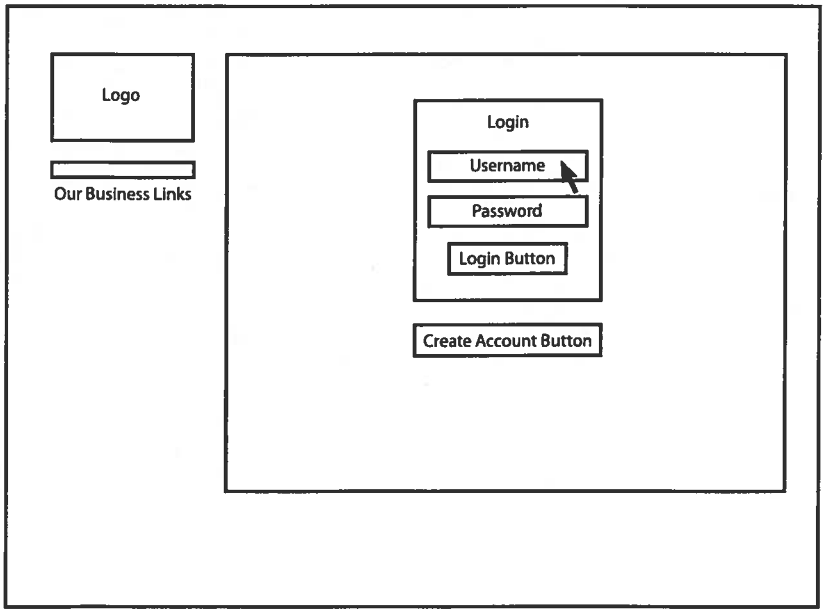
Gator Market Use Case: Seller

3) Listing Page



The listing page shows the details of the Listing such as Price, Description, and Images. User clicks the Buy Button, but is not logged in yet. So they are first taken to the Login Page.

4) Login Page



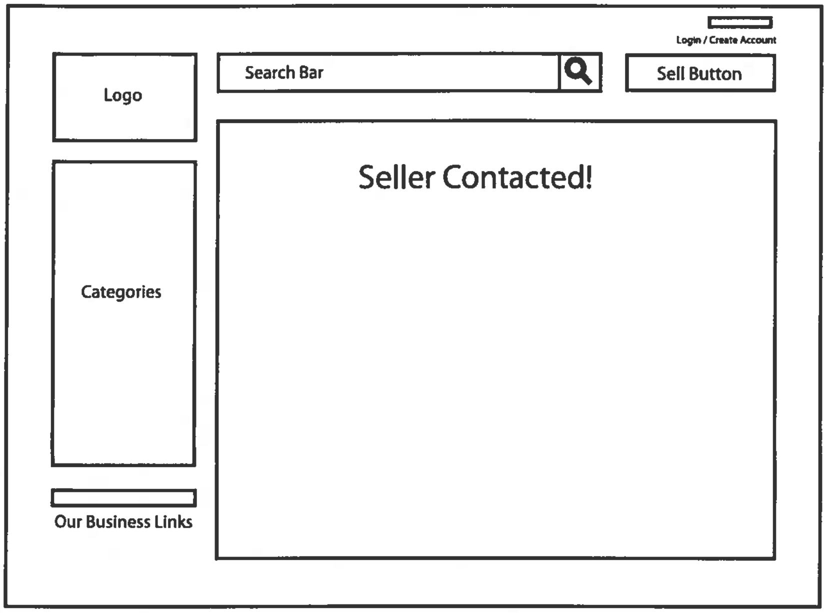
The Login Page allows a Registered User to log into their account, or an Unrestistered User to create a new account.

The user fills out the fields Username and Password, and then clicks the login button, and then Buy interaction will be completed if the login credientials are correct.

Group 2

Gator Market Use Case: Seller

5) Seller Contacted Page

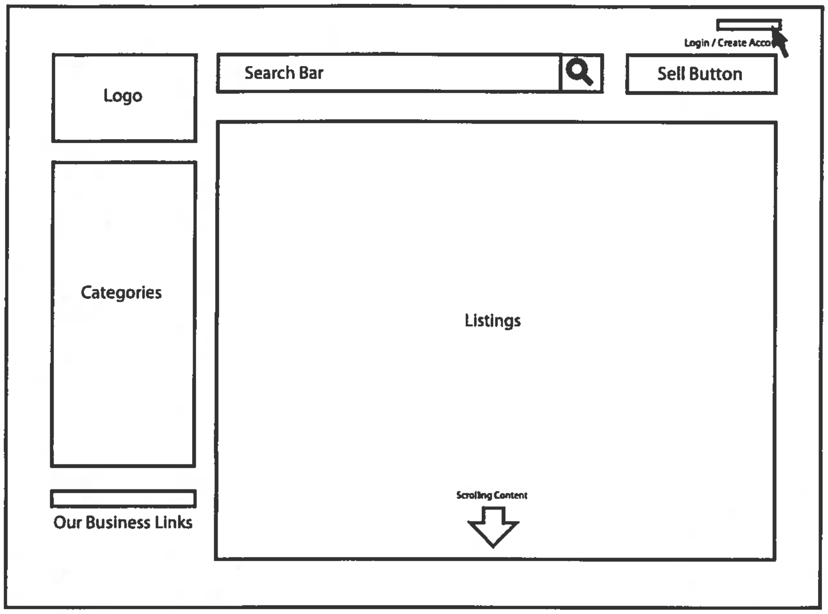


After the user clicks the buy button and finishes logging in, the seller is contacted through the site and the user is taken to a success page.

Group 2

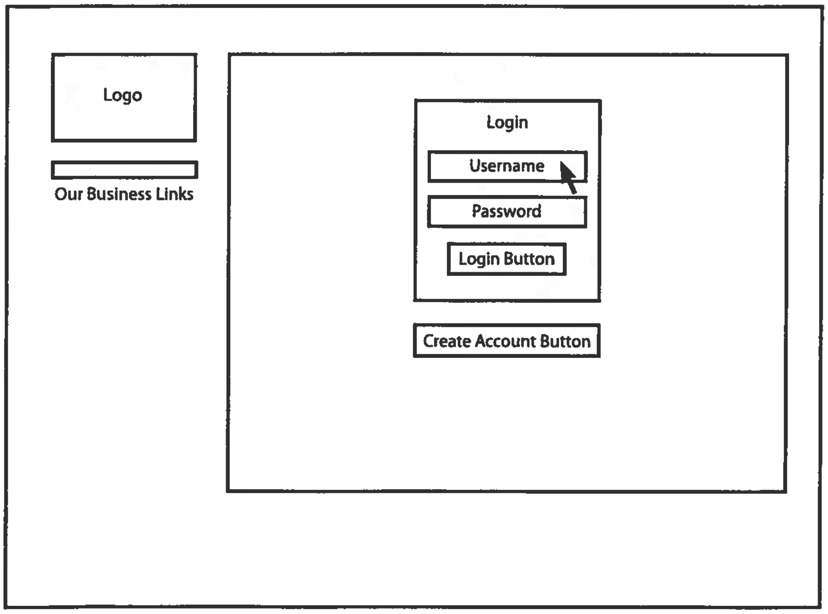
Gatorr Market Use Case: Seller

1) Landing Page



User arrives at the landing page, and then logs in because they are a registered user that wishes to sell an item.

2) Login Page



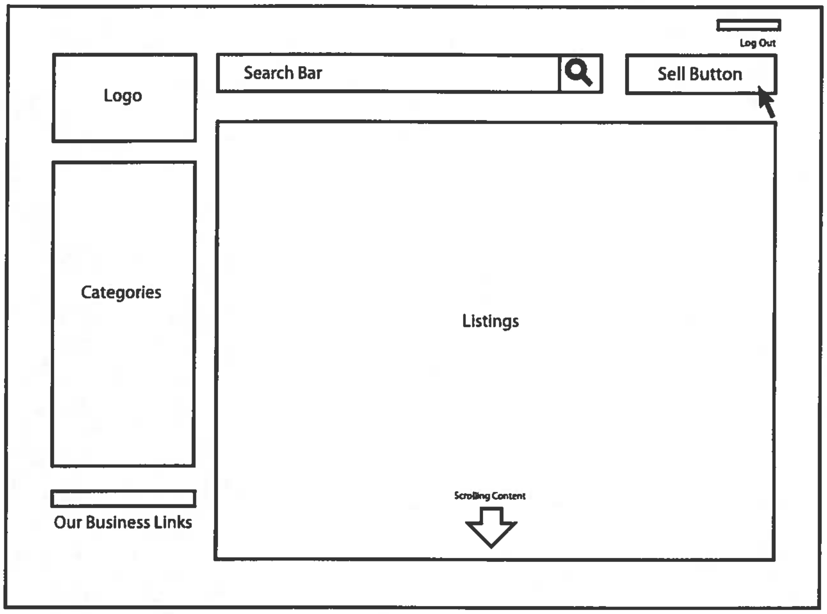
The Login Page allows a Registered User to log into their account, or an Unrestistered User to create a new account.

The user fills out the fields Username and Password, and then clicks the login button, and then they are taken back to the landing page.

Group 2

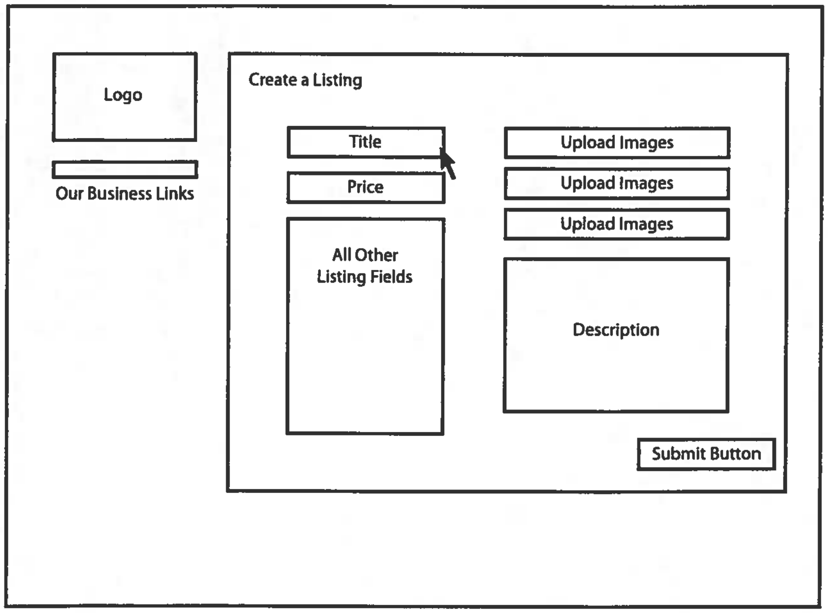
Gator Market Use Case: Seller

3) Landing Page - Sell Button



The user clicks the sell button and is taken to the Listing Creation page.

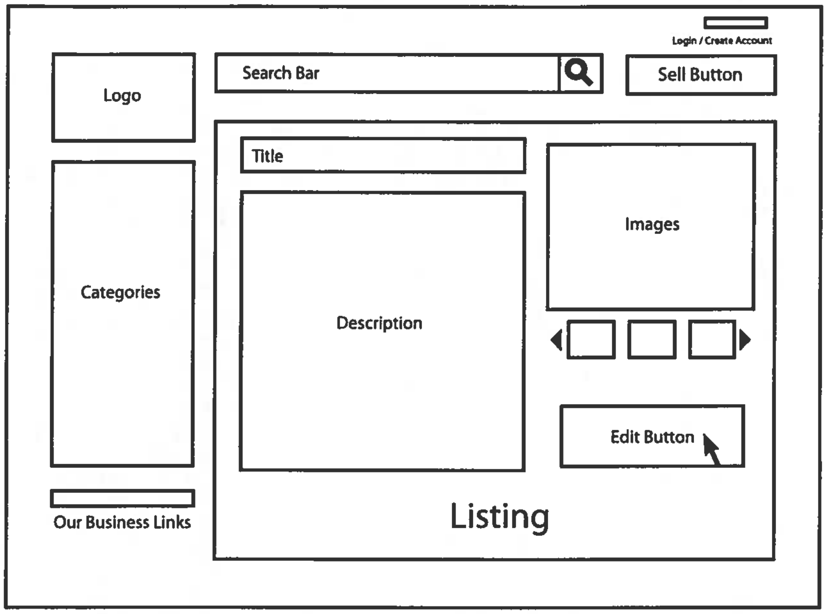
4) Listing Creation Page



The Listing Creation Page allows a Registered User to create and post a listing to the site.

The user fills out the fields with the relevent information about the item they wish to sell and clicks the submit button.

3) Listing Page

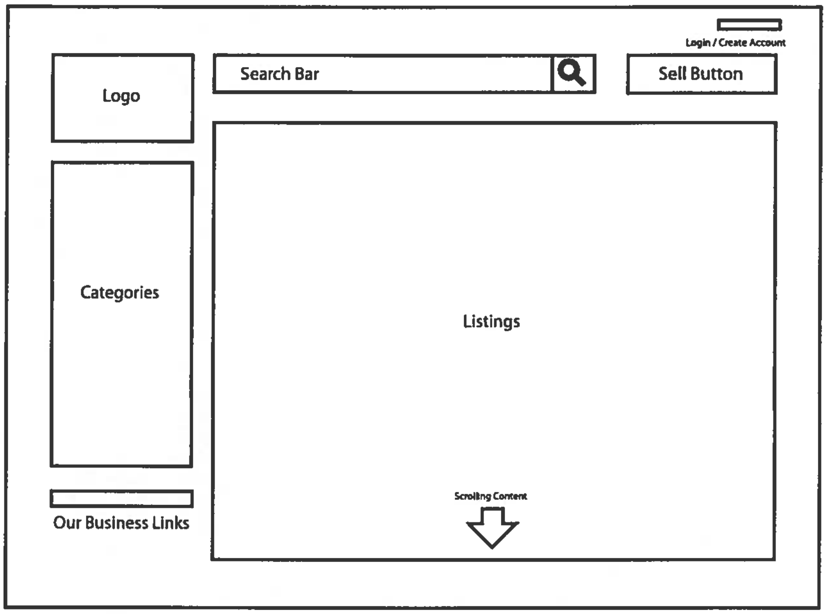


After the Seller submits a Listing, they will be taken to the Listing page where they can review the information and edit the Listing further if needed by clicking the Edit button, which will take the user back to the Listing Creation Page where the Seller can edit the fields which will now be populated with the information they submited earlier.

Group 2

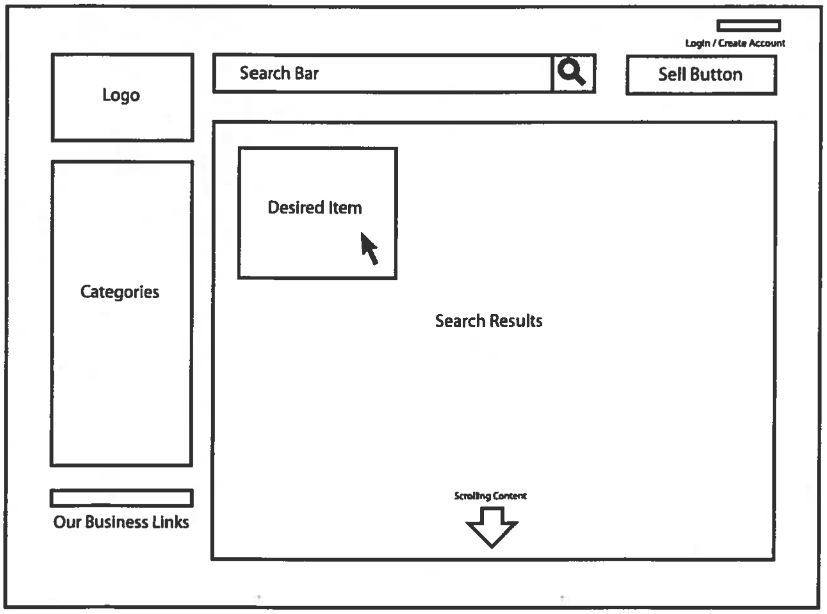
Gator Market Use Case: Unregistered User

1) Landing Page



An Unregistered user arrives at the landing page, and then browses the list of categories on the left.

2) Category Results

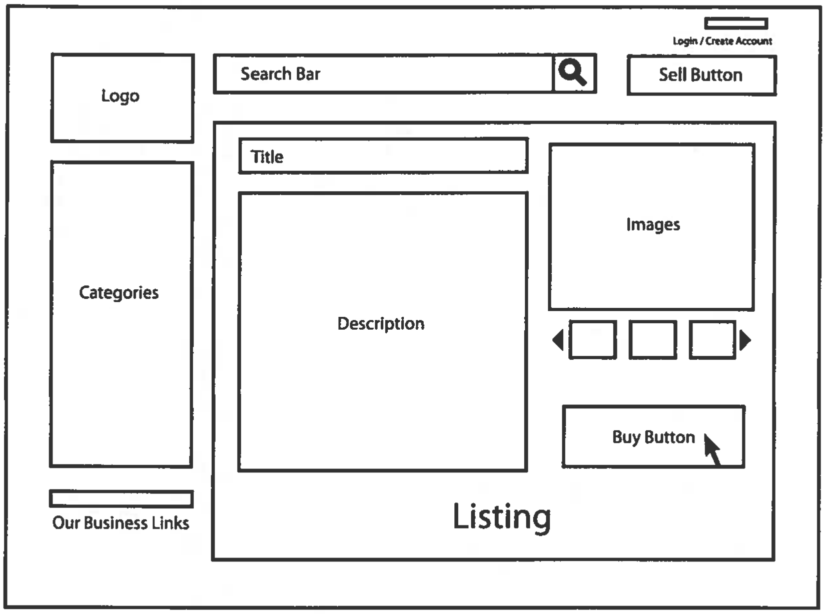


The User clicks a category button and is then taken to a page that shows relevant Listings marked with that Category Tag. The user then clicks on the desired Listing and is taken to the Listing page.

Group 2

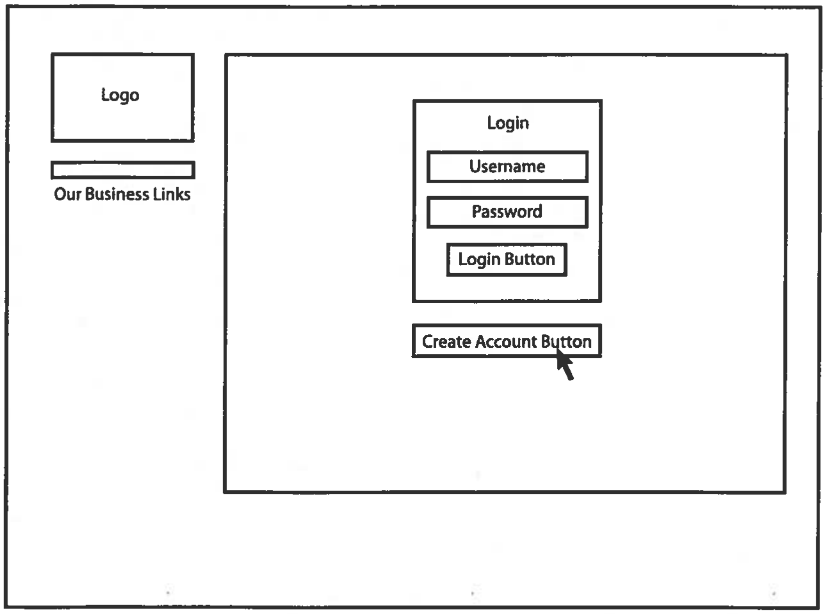
Gator Market Use Case: Unregistered User

3) Listing Page



The listing page shows the details of the Listing such as Price, Description, and Images. User clicks the Buy Button, but is not a Registered User. So they are first taken to the Login Page.

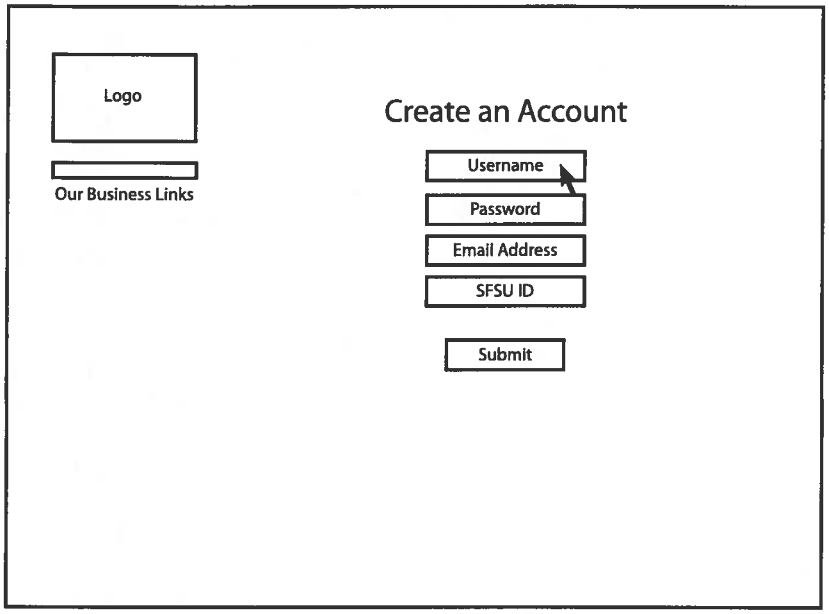
4) Login Page



The Login Page allows a Registered User to log into their account, or an Unrestistered User to create a new account.

The User does not have an account so they click the Create an Account Button.

4) Create an Account Page



The Create an Account Page contains fields that a User must fill out in order to create an account.

The User fills out the appropriate information and after clicking the submit button, they shall be taken to an Account Creation Success page with links that shall allow them to return to the Landing Page or the page that brought them to the Create An Account Page.

7. High-Level System Architecture & Database Organization

Software Products and Components

- Linux - Operating System that provides a base foundation on our server instance for all the other systems to run. We are running Ubuntu 16.04
- Apache - Web server software that handles user interaction and fetches requests from the server. For example when the user goes to the home page
- MySQL - Database to hold information. We will be using our database to store login information about the user as well as the products we will be listing
- PHP - Used for generating dynamic web pages. When the user makes a request that requires PHP, Apache sends the request to PHP and waits for it to generate an HTML file to send back to the user. For example when the user logs onto our webpage we will need to make PHP calls to the database to check to see if the user is there and generate the correct page based on their action.
- Model-View-Controller - We will be using the MVC framework which is divided into three parts, Model, View and Controller. The View is what is actually displayed on your screen when you view a website, and it changes based on the Model. The Model collects and stores information that the Controller instructs. And the Controller gets information from the user and updates the the Model.

Database Organization

- User
    - **Username** - varchar(25)
    - **Student ID Number** - int(9)
    - **Password** - varchar(25)
    - **Email -** varchar(40)
    - **Agree to EULA -** tinyint(1)
    - **Admin Status -** tinyint(1)
- Listings
    - **Student ID Number -** int(9)
    - **Title -** varchar(40)
    - **Price -** decimal(7,2)
    - **Image -** BLOB
    - **Description -** varchar(300)
    - **DescriptiveTag** - varchar(15)

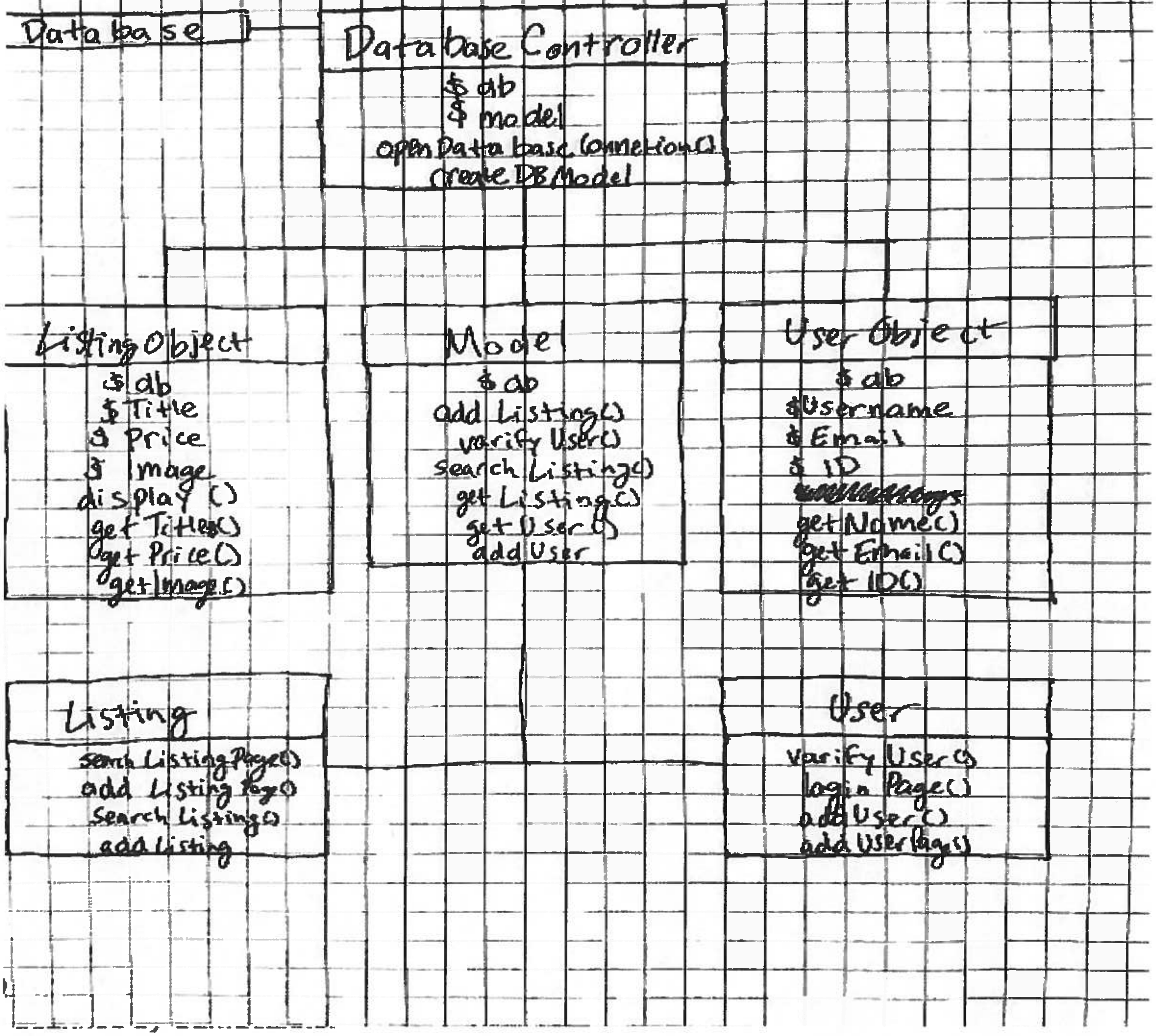
Search Algorithm

SELECT * FROM Listing WHERE DescriptiveTag = 'tag' AND Description LIKE 'searchkey';

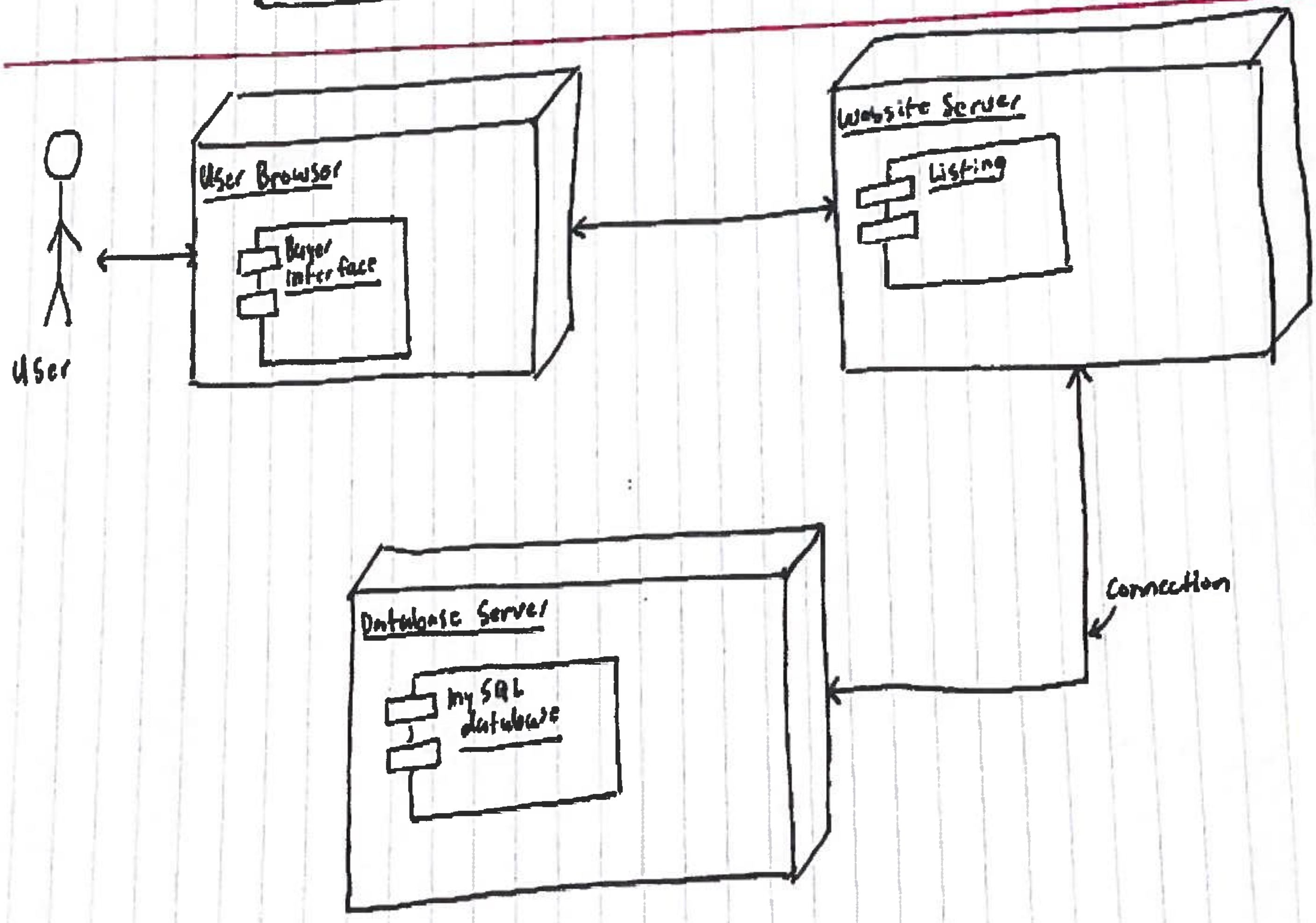
- Conditions
    - If user does not select category, default Descriptive Tag is title
    - If searching just by name we will search in both the title and description

8. High-Level UML Diagrams

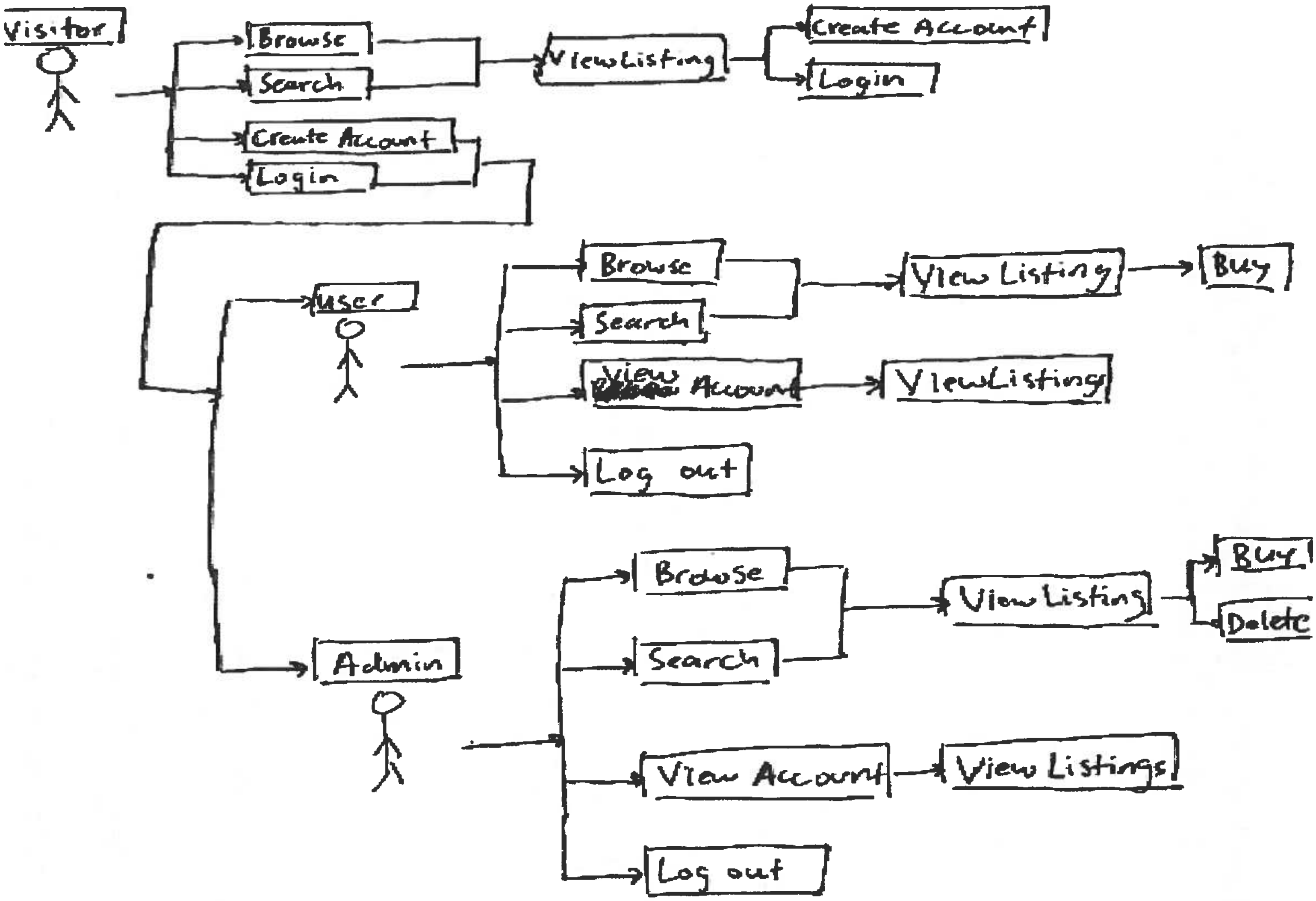
UML CLASS DIAGRAM



Components and Deployments Diagram



Use Cases Diagram



9. High Level API

DatabaseController
Datafield:

- **$db** - holds and maintains the connection to the database
- **$model** - used to create a Model Object to call functions in the controller

Functions:

- **openDatabaseConnection()** - creates PDO connection to the database
- **createDBModel()** - sets $model to be a new object of type Model

Model
Datafield:

- **$db** - holds and maintains the connection to the database

Functions:

- **addListing($title, $price, $image)** - add listing to the database with the specific input values
- **varifyUser($id, $password)** - checks the database to see if there is a matching ID and Password combination; returns boolean
- **searchListing($keyword) -** searches database for title that is similar to keyword; returns PDO that contains results
- **getListing($PDO)** - takes in the PDO results and returns an array of ListingObjects
- **getUser($id) -** searches the database for the Use that matches the ID; return UserObject
- **addUser($name, $id, $email) -** adds new user to database

ListingObject
Datafield:

- **$db** - holds and maintains the connection to the database
- **$Title -** name of the item for sale
- **$Price -** price of item for sale
- **$Image -** image of itme for sale

Functions:

- **display() -** displays the title, price and image of the ListingObject
- **getTitle()** - returns the title
- **getPrice()** - returns the price
- **getImage()** - returns the image

UserObject
Datafield:

- **$db** - holds and maintains the connection to the database

- **$Name -** username
- **$Email -** email of the user
- **$ID -** id of the user

Functions:

- **display() -** displays the title, price and image of the ListingObject
- **getName()** - returns the username
- **getEmail()** - returns the email
- **getID()** - returns the user ID

Listing

Functions:

- **searchListingPagie() -** sets up the views for the listing page
- **addListingPage()** - sets up the views for the add listing page
- **searchListing()** - gets called when the Submit button on the searchListing page is pressed, it takes in the values from the text fields and sends it to the Model searchListing, creates ListingObjects, then displays the objects
- **addListing()** - gets called when the Submit button on the addListing page is pressed, it takes in the values from the text fields and sends it to the Model addListing function

User

Functions:

- **createAccountPage() -** sets up the views for the create account page
- **loginPage()** - sets up the views for the login page
- **varifyUser() -** gets called when the submit button is pressed on the loginPage, takes in the values from the text fields and send it to the Model varifyUser function
- **addUser() -** gets called when the submit button is pressed on the createAcountPage, takes in the values from the text fields and send it to the Model addUser function

10. Key Risk

	Key Risks	Response
Skill Risks	● Team members may not be proficient in PHP. ● Team members have no prior experience in MVC.	● Team will work together and learn together in order to complete the project.
Schedule Risks	● Working schedules conflict with members may cause members to not be able to work together as a team.	● Team will work together on days that everyone could meet and work on their own in their individual time.
Teamwork Risks	● Communication	● Will use email.

11. **Team Roles**

- Jeffrey Noehren
  - Team Lead
  - Backend development
- Jason Newman
  - Tech Lead
  - Frontend development
- Kenny Tran
  - Backend development
- Olatope Oladipupo
  - Frontend development
- Maria Lienkaemper
  - Backend development

SW Engineering CSC 648/848 Summer 2016
Gator Market

Group #2 (The Fighting Mongooses)
Local
Jeffrey Noehren - jeffreynoehren9@gmail.com
Jason Newman
Olatope Oladipupo
Kenny Tran
Maria Lienkaemper

“Milestone 3 feedback summary document”
07/31/16

Summary of the feedback:

For our team meeting with Professor Dragutin Petkovic and Professor Marc Sosnick, we went over the review of our prototypes and planning for the final product delivery. During this scheduled 30 minute review meeting, we gone over important objectives of milestone three such as defining what kind of product we're planning to deliver, ensuring that the software development process is being done on a timely manner, feedback on the major UI screens, software architecture, algorithms, identification of technical risks, teamwork, and effectiveness of software development.

In the reviewing process of our prototype website, *Gator Market,* Professor Petkovic recreated and went through a simulation of pretending to be a user accessing the site from his own home. We were not allowed to interfere with his process of reviewing with instructional comments except for answering technical questions asked by him. Upon Professor Petkovic's first glance at the product, he was able to identify that this was a site for San Francisco State University students to buy and sell a variety of things. However, since our website only has the title of our site, we were suggested to add some text or a short one line description below such as, "Welcome to Gator Market, where you can buy and sell things!" to add more clarity for users who first enter the website.

Professor Petkovic took note that that our software development was on the right track as we had all the necessary SW components installed and integrated correctly in our vertical prototype. He told us that for our prototype, "all the elements are here." As of this meeting, he told us not to discard what we worked on and that we will continue to work with the current code to quickly bring it up to become the final delivery code which would in return become a presentable product to our targeted audience.

For our major UI screens, Professor Petkovic told us that there were a lot of empty and blank spaces in some of our screens such as the listing screen that displays our listings. To fix this, we already have a plan in development to address that issue by resizing the images to fit properly on the screen so that there would not be that big white space above. The next issue that was brought up was the navigation bar for our listing categories. Our navigation bar was not aligned the same length which gave it more of an oddly shaped style. It was suggested that we correct that by making them all the same length so that the navigation bar looks cleaner. Another issue is that the placement of some things are weirdly placed. Professor Petkovic recommended that we move a few things to fit the traditional convention of a buy and sell website such as “Amazon”. He also told that we should possibly include another search bar at the very top of the website and design some pull-down menus for search categories. A couple of key things that we were told missing in our website was our google analytics page, shopping cart, number of listing results shown, and standard contact/policy information at the bottom of the website. Although all of that, we have a great search bar and our listings page included things such as the the image, title, and price of the listing.

Our software architecture has no signs of problems as noted by Professor Petkovic. The database and overall design is doing well. He also told us that our searching algorithm was doing fine and that it was nice and simple. He was able to search ‘books’ and find results of books being returned to him. Overall, Professor Petkovic said that big thing about the website in its current state is that the general layout needs to be redesigned because it wastes space, feels bumpy, and has shows to make blank spaces. The good thing is that we have all the necessary elements of the website and that our searching algorithm is working great.

After the website reviewing process with Professor Petkovic, both Petkovic and Sosnick identified and addressed the technical risks with our team. A few things that they pointed out was that our number of commits for GitLab had a largely uneven distribution, but that was resolved through the different types of tasks that members were assigned to do. Our teamwork and software development was evaluated in terms of effectiveness and we were able to keep our overall productivity and effectiveness high for everyone.

Priority 1 functionality:

Priority 1

1. Users shall be taken to the landing page upon entering the site.

2. Search bar shall be present on each page of the website for users to search listings.

3. Unregistered Users

    3.1 Unregistered Users shall only be able to browse and search for listings.

    3.2 Unregistered Users shall only be able to view the following:

        3.2.1 Name of listing for sale.

        3.2.2 Price

        3.2.3 Image of listing for sale.

    3.3 Unregistered Users shall be able to create a User Account with a valid SFSU ID.

4. Registered Users

    4.1 Registered Users shall be able to browse and search for listings.

    4.2 Registered Users shall be able to manage their User Account.

    4.3 Registered Users shall be able to post a listing that includes the following:

        4.3.1 Username

        4.3.2 Name of listing for sale

        4.3.3 Price

        4.3.4 Image(s)

        4.3.5 Posting Date

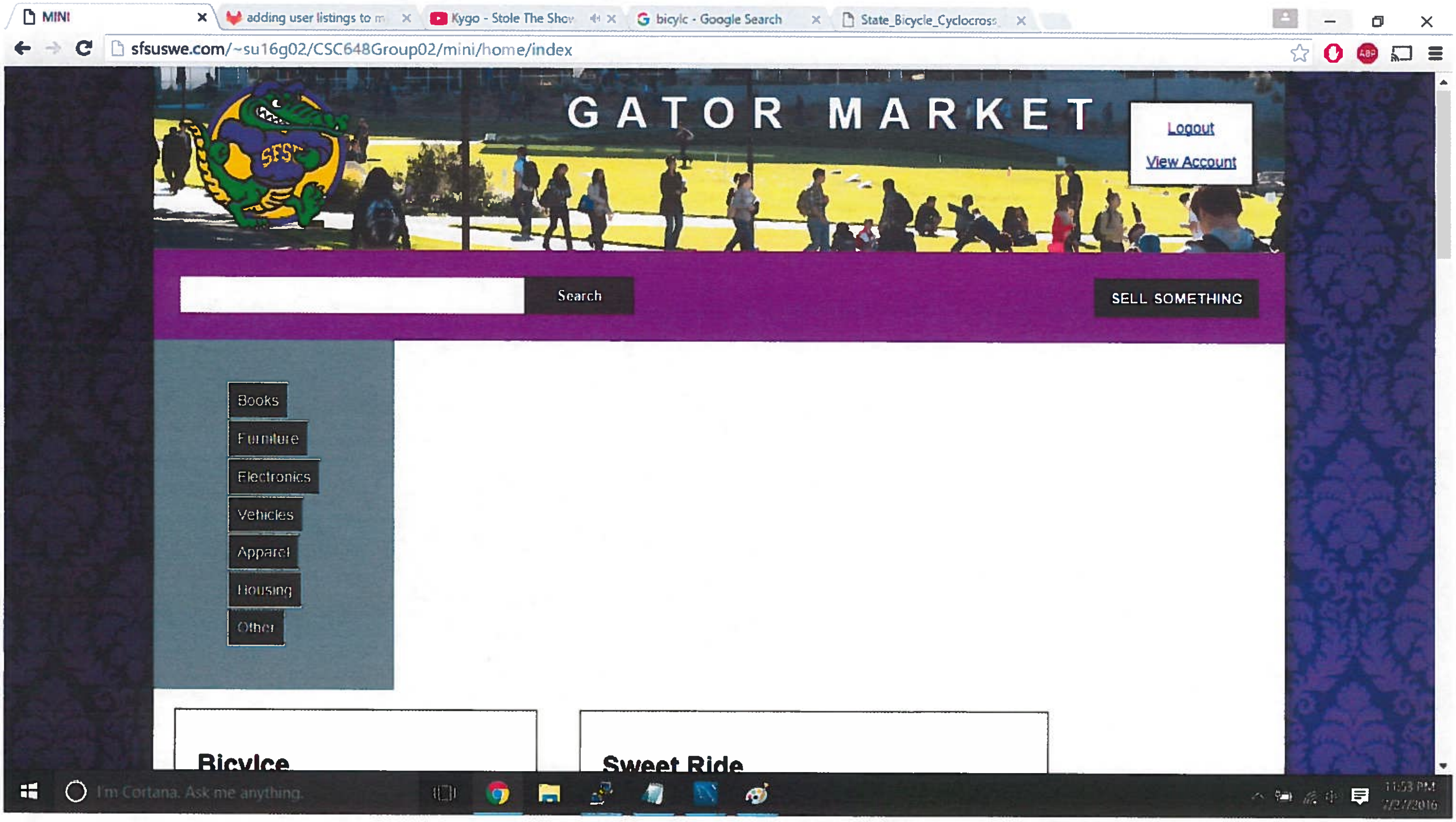
        4.4.6 Description

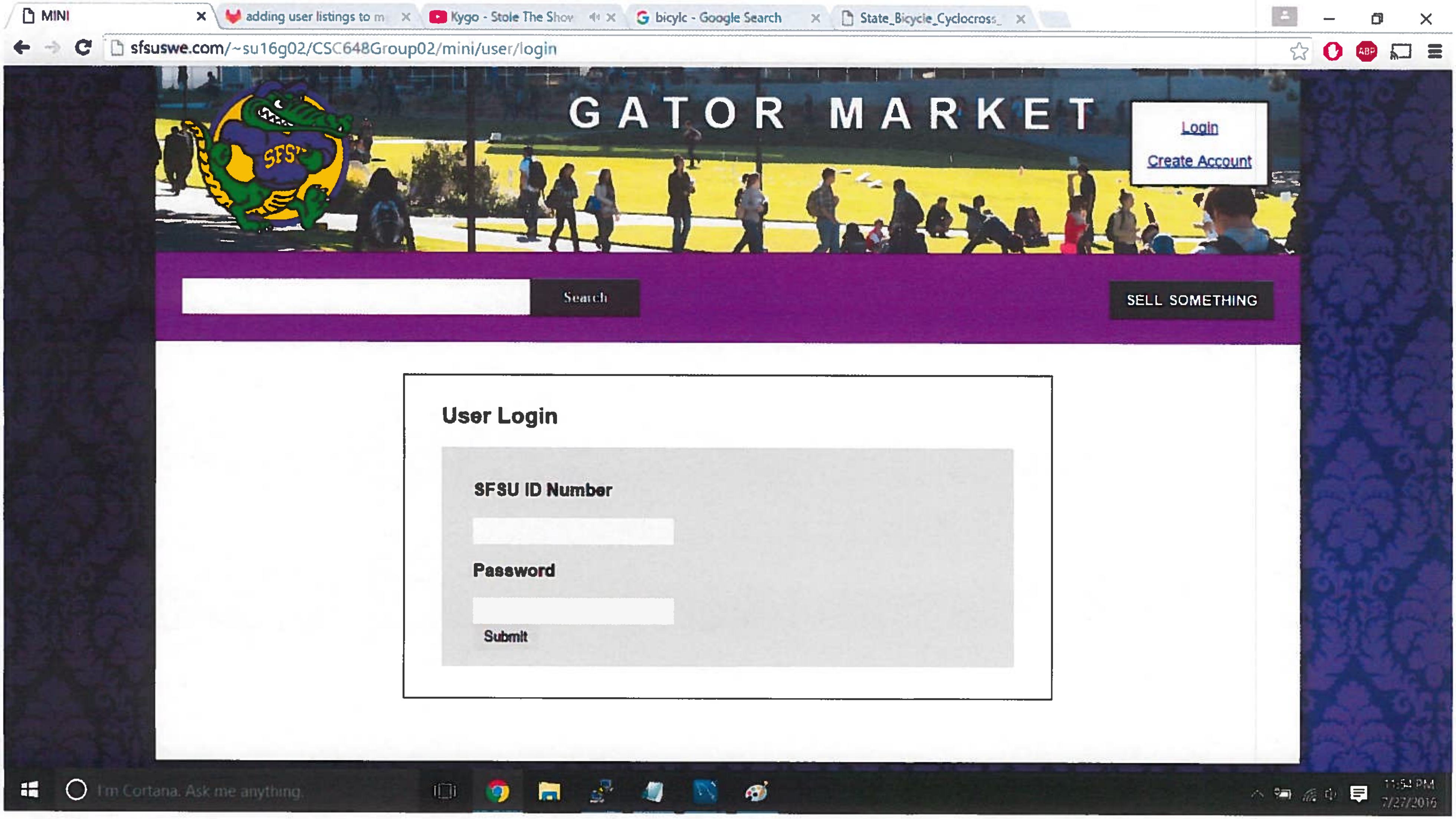
5. Administrators

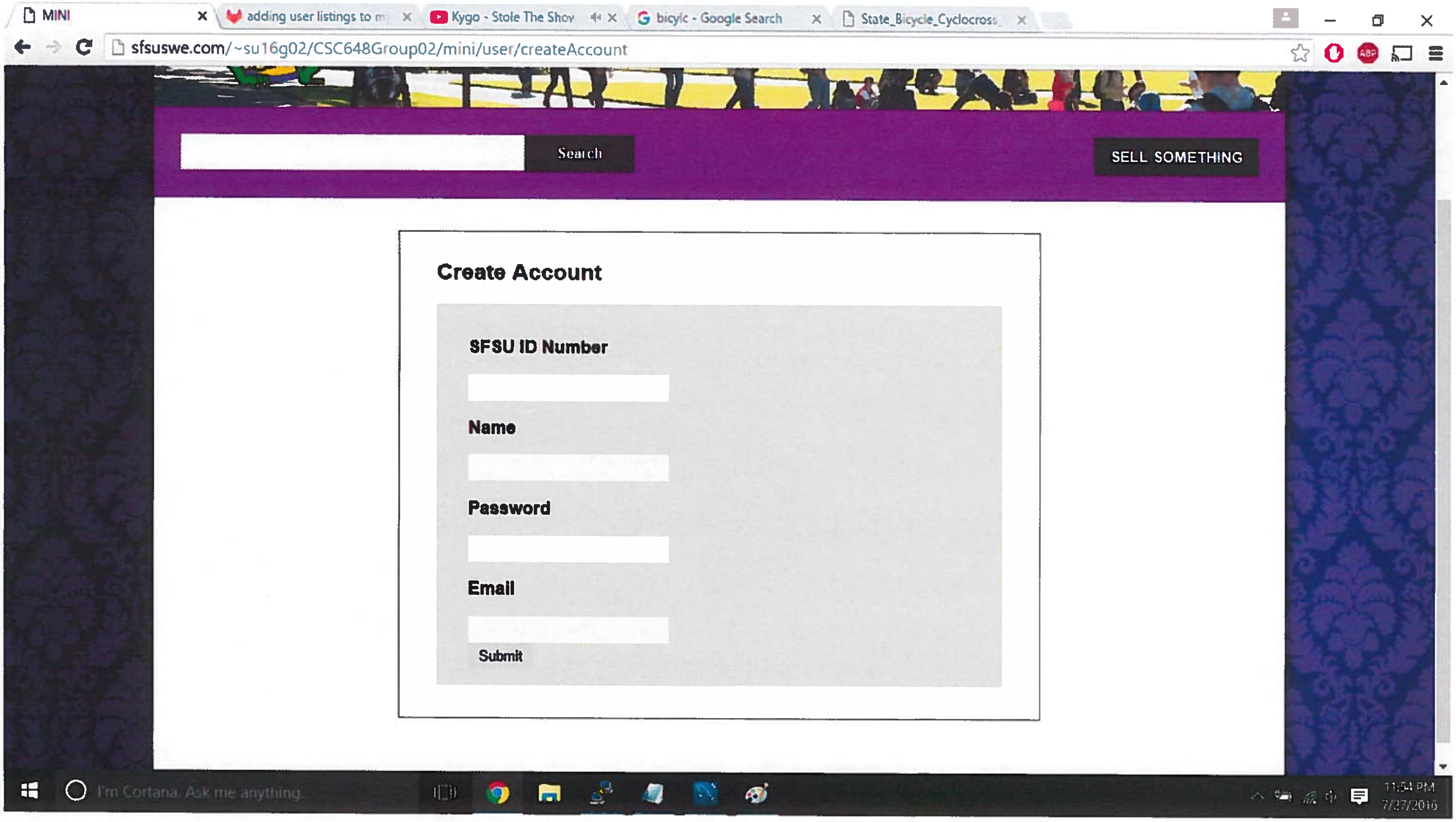
    5.1 Administrators shall be able to manage all users registered to the site.

    5.2 Administrators shall be able to delete listings in violation of the EULA.

        5.2.1 Registered Users shall be notified with a reason as to why listing was deleted.


MINI
adding user listings to m
Kygo - Stole The Show
bicylc - Google Search
State_Bicycle_Cyclocross
sfsuswe.com/~su16g02/CSC648Group02/mini/home/index
GATOR MARKET
SFSU
Logout
View Account
Search
SELL SOMETHING
Books
Furniture
Electronics
Vehicles
Apparel
Housing
Other
Bicylce
Sweet Ride
I'm Cortana. Ask me anything.
11:53 PM
7/27/2016


MINI
adding user listings to m
Kygo - Stole The Show
bicylc - Google Search
State_Bicycle_Cyclocross_
sfsuswe.com/~su16g02/CSC648Group02/mini/user/login
SFSU
GATOR MARKET
Login
Create Account
Search
SELL SOMETHING
User Login
SFSU ID Number
Password
Submit
I'm Cortana. Ask me anything.
11:54 PM
7/27/2016


MINI
adding user listings to m
Kygo - Stole The Shov
bicylc - Google Search
State_Bicycle_Cyclocross
sfsuswe.com/~su16g02/CSC648Group02/mini/user/createAccount
Search
SELL SOMETHING
I'm Cortana. Ask me anything.
11:54 PM
7/27/2016

Create Account

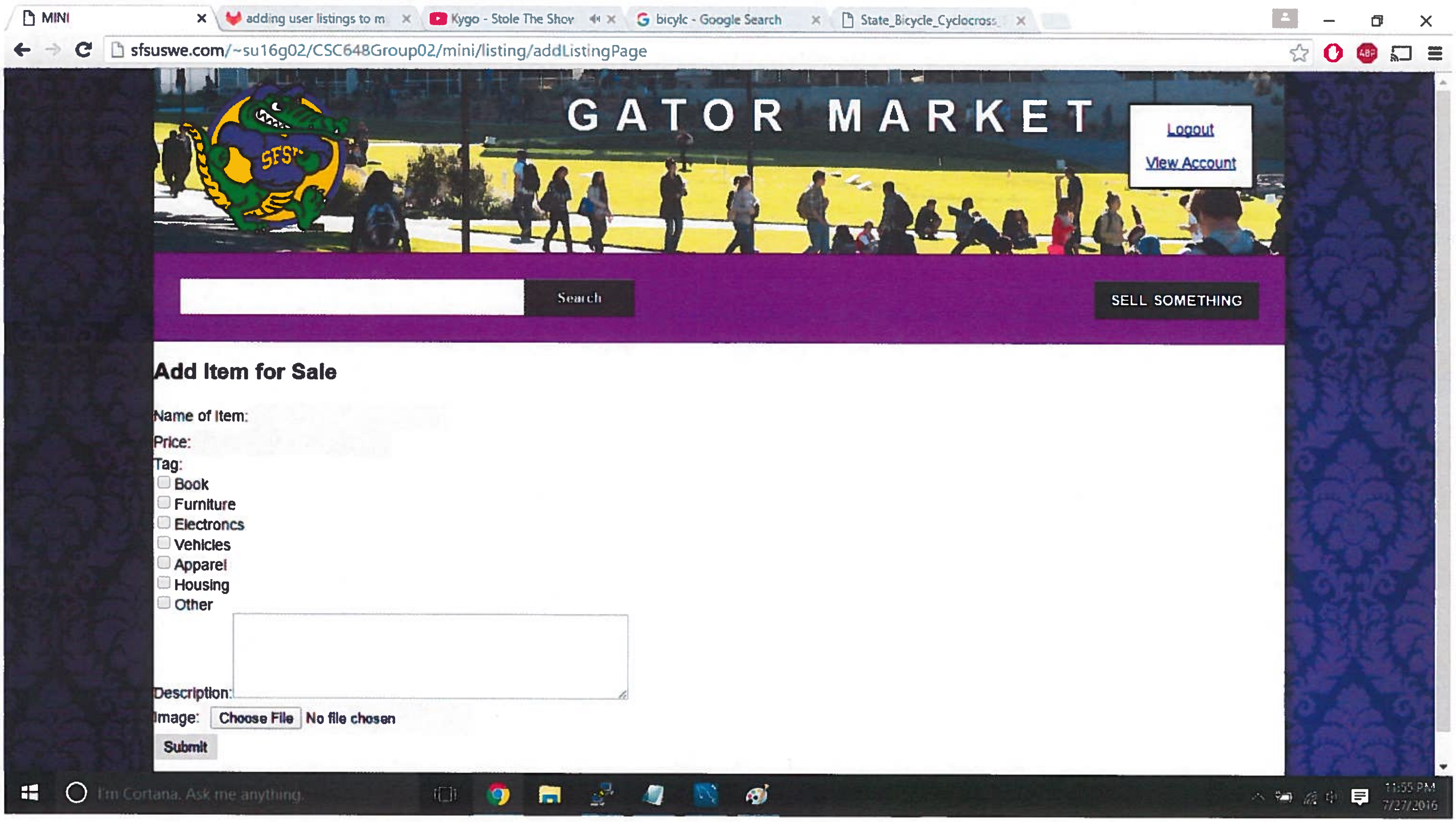
SFSU ID Number

Name

Password

Email

Submit



Add Item for Sale

Name of Item:

Price:

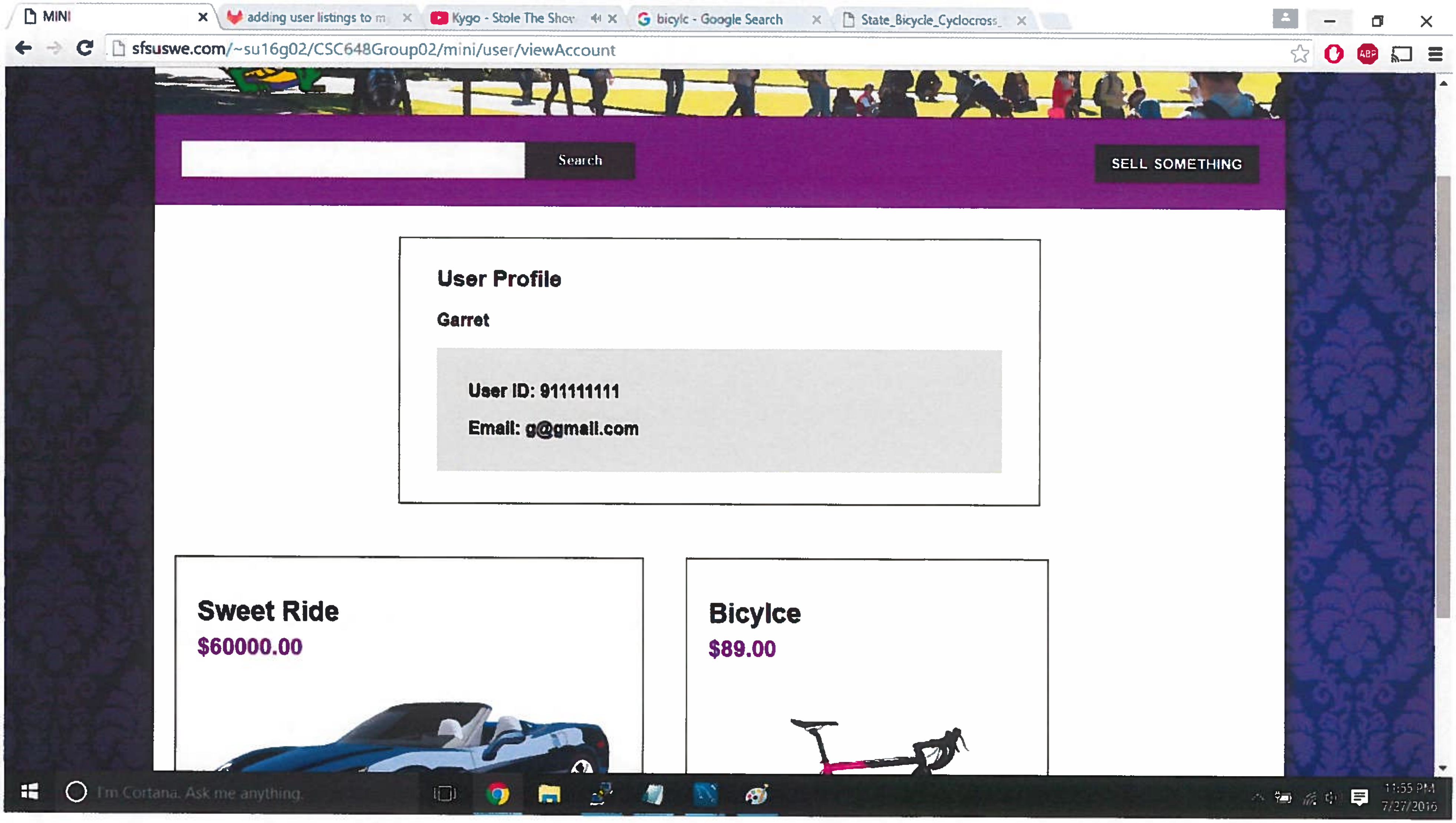
Tag:

- ☐ Book
- ☐ Furniture
- ☐ Electroncs
- ☐ Vehicles
- ☐ Apparel
- ☐ Housing
- ☐ Other

Description:

Image: Choose File No file chosen

Submit



MINI
adding user listings to m
Kygo - Stole The Show
bicylc - Google Search
State_Bicycle_Cyclocross_
sfsuswe.com/~su16g02/CSC648Group02/mini/user/viewAccount
Search
SELL SOMETHING
I'm Cortana. Ask me anything.
11:55 PM
7/27/2016

User Profile

Garret

User ID: 911111111

Email: g@gmail.com

Sweet Ride

$60000.00

Bicylce

$89.00

SW Engineering CSC 648/848 Summer 2016
Gator Market

Group #2 (The Fighting Mongooses)

Local

Jeffrey Noehren - jeffreynoehren9@gmail.com

Jason Newman

Olatope Oladipupo

Kenny Tran

Maria Lienkaemper

"Milestone 4"

08/09/16

History Table

Revision #	**Date**
1	08/05/16
2	08/09/16

Table of Contents

Project Summary

Our product, Gator Market is a simple and easy-to-use online marketplace designed for San Francisco State University (SFSU) students to buy and sell goods with each other. Upon accessing the website, users will be taken to the home landing page where they will be presented with a short description of what the website is about and image postings of listings that students are selling. This gives users who visit to get an overall idea of the website and also give them something to nice to see so that they will start to browse through listings to spark their interest into buying things right away.

A user may start browsing through listings through clicking the 'next' button below the pages to browse the the most recent postings or they may take advantage of the search bar up top which allows simple searching of listings and categorizing capabilities through the drop down menu on the side.

Registration is very simple, it only requires a user to fill out a simple form including their student ID, username, password, and valid SFSU e-mail. A registered user is able to do anything that an unregistered user can do as well as having the options to buy and sell listings on the website. They are able to view more details on their listings made by other students such as name of the seller, posting date, and description of listing. Registered users can also view their account to view details what listings they are selling. An unregistered user only has the ability to browse and search for listings and view the basic details of it which are names of the listings for sale, prices, and images. If an unregistered user attempts to purchase and sell something, they will be taken to the login page where they will be prompted to fill in their login credentials.

Purchasing products is very easy and intuitive for prospective buyers, there are big buttons throughout that display, "Buy". There is one on the front that displays products and then another one that is displayed when users click the products to view more details about it. This ensures visibility and a marketing strategy that tempts the users into wanting to buy our products.

Although this website is currently only catering to SFSU students, the website is publicly visible for anyone to view it regardless if whether they are actually a SFSU student or not. This is a good way to increase people traffic by letting people look and browse the website unlike other websites out places restrictions such on viewing unless they are registered. There is much potential to this website and up for possible ideas to be extended in the future to include all types of schools.

Gator Market provides the following functions:

- The ability for users to create an account with Gator Market
- Browse, search, and view products
- Post products to Gator Market for sale
- Categories in the search bar

- Ability to purchase and contact seller

URL: http://sfsuswe.com/~su16g02/CSC648Group02/mini/

Usability Test Plan

Test Objective:

The objective of the usability test for the Sell Something feature of our website, Gator Market is to ensure that the process is simple and easy to use. We feel that to be able to easily sell something on Gator Market is essential for the project due to the fact that this website was created as a place where San Francisco State Students are able to buy and sell things to other SF State students. If the Sell Something feature is flawed, the whole website will be flawed.

Test Plan:

System Setup:

Starting Point: The tester will be given proper login information and a URL for the home page of the website.

Task to be Accomplished: The tester shall be able to upload an item for sale on Gator Market

Completion Criteria: The user shall post an item to sell easily on Gator Market.

Intended User: SFSU students who are looking to buy or sell used goods in a central online location.

URL: http://sfsuswe.com/~su16g02/CSC648Group02/mini/

Questionnaire:

Question:	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
The sell something button is clearly visible					
The Login Page is easy to find					
The Sell Something Page has a good layout					
Selling an item on Gator Market is easy					
I can find my item for sale easily					

Comments	

Quality Assurance Test Plan

Test Objective:

This Quality Assurance test is to make sure that the Sell Something feature is designed and working properly. This is not designed to test usability.

HW and SW Setup: The website shall be viewable in a standard desktop/laptop browser, and shall render correctly on the latest version of all major browsers: Mozilla Firefox, Google Chrome, Safari, and Internet Explorer.

Feature to be Tested: Sell Something

Actual Test Cases:

Number	Description	Test Input	Expected Output	Pass/Fail
1.	Testing button	Clicking on “Sell Something” button not logged in	Taken to login in page	Pass
		Clicking on “Sell Something” button while logged in	Taken to “Sell Something” Page	Pass
2.	Testing input fields	Upload something to sell with input in all fields	Item is successfully uploaded	Pass
		Upload something to sell with input in all fields except a description	Item is successfully uploaded	Pass
		Upload something to sell with input in all fields except a tag	Item will not be uploaded	Pass
		Upload something to sell with input in all fields except a price	Item will not be uploaded	Pass
		Upload something to sell with input in all fields except a image	Item will not be uploaded	Pass
		Upload something to sell with input in all fields except a title	Item will not be uploaded	Pass

Code Review

Sell Something

M4 Code Review

3 messages

---

jeffrey noehren <jeffreynoehren9@gmail.com> Thu, Aug 4, 2016 at 3:43 PM

To: Maria Lienkaemper <mlienkae@mail.sfsu.edu>

View:

```
<div class="container">
  <div class="center_box">
  <div class="grey_box">
  <div class="left_align">
  <form action="<?php echo URL;?>listing/addListing"  method="post"
enctype="multipart/form-data">
    <h2>Add Item for Sale</h2>
    <h4>Name of Item:</h4><input type="text" name="title" maxlength="30"><br />
    <h4>Price:</h4><input type="text" name="price" maxlength="7"><br />
    <h4>Tag:</h4><br>
    <input type="checkbox" class="radio" name="tag" value="book">Book<br>
    <input type="checkbox" class="radio" name="tag" value="furniture">Furniture<br>
    <input type="checkbox" class="radio" name="tag"
value="electronics">Electroncs<br>
    <input type="checkbox" class="radio" name="tag" value="vehicles">Vehicles<br>
    <input type="checkbox" class="radio" name="tag" value="apparel">Apparel<br>
    <input type="checkbox" class="radio" name="tag" value="housing">Housing<br>
    <input type="checkbox" class="radio" name="tag" value="other">Other<br>

    <h4>Description:</h4><textarea name="description" maxlength="500" cols="50" rows
="5"></textarea><br />
    <h4>Image:</h4><input type="file" name="image"><br />
    <input type="hidden" name="sellerID" value="<?php echo
```

```
$_SESSION['user']->getID(); ?>">
    </div>
    <input class="submit_button" type="submit" name="submit">
  </form>
  </div>
  </div>
</div>
```

Controller:

```
public function addListing()
    {
      if(isset($_POST['submit']))
      {
        $blob = file_get_contents($_FILES['image']['tmp_name']);
        $imageType = $_FILES['image']['type'];
          if(substr($imageType,0,5) == "image")
          {
            $res =$this->model->addListing($_POST['title'], $_POST['price'], $blob,
$_POST['description'], $_POST['tag'], $_POST['sellerID']) ;
          }
      }
      header('location: ' . URL . 'listing/addListingPage');
      if($res)
      {
        echo"your listing has been uploaded";
      }
      else{echo "we could not upload your listing";}
    }
```

Model:

```
public function addListing($title, $price, $image, $description, $tag, $sellerID)
    {
      //print_r($_SESSION['userID']);
      $sql = "INSERT INTO listing(title, price, image, description, tag, sellerID)
VALUES(:title, :price, :image, :description, :tag, :sellerID)";
      $query = $this->db->prepare($sql);
      $parameters = array(':title' => $title,':price' => $price, ':image' => $image,
'description' => $description, ':tag' => $tag, ':sellerID' => $sellerID);
      $query->execute($parameters);
    }
```

---

jeffrey noehren <jeffreynoehren9@gmail.com> Thu, Aug 4, 2016 at 3:45 PM

To: Maria Lienkaemper <mlienkae@mail.sfsu.edu>

Hey Maria,

I know I said that you should just focus on the presentation this weekend but we still need someone to do code review for our M4 document and I think you would probably do this best. All you have to do is go over this code and make sure we are following our coding style and make comments on things that need to be fixed! It shouldnt take you too long honestly but tell me if you have any questions!

Jeffrey
[Quoted text hidden]

Maria Lienkaemper <mlienkae@mail.sfsu.edu> Fri, Aug 5, 2016 at 6:18 PM

To: jeffrey noehren <jeffreynoehren9@gmail.com>, Maria Lienkaemper <mlienkae@mail.sfsu.edu>

View:

Tabs are inconsistent with the agreed style, they are two spaces instead of four.

```
<div class="container">
```

The div class names are descriptive and well chosen.

```
  <div class="center_box">
```

```
 <div class="grey_box">
 <div class="left_align">
 <form action="<?php echo URL; ?>listing/addListing"  method="post"
enctype="multipart/form-data">
   <h2>Add Item for Sale</h2>
   <h4>Name of Item:</h4><input type="text" name="title" maxlength="30"><br />
   <h4>Price:</h4><input type="text" name="price" maxlength="7"><br />
   <h4>Tag:</h4><br>
   <input type="checkbox" class="radio" name="tag" value="book">Book<br>
   <input type="checkbox" class="radio" name="tag" value="furniture">Furniture<br>
```

Label for the Electronics category is misspelled in the tag options.

```
   <input type="checkbox" class="radio" name="tag" value="electronics">Electroncs<br>
   <input type="checkbox" class="radio" name="tag" value="vehicles">Vehicles<br>
   <input type="checkbox" class="radio" name="tag" value="apparel">Apparel<br>
   <input type="checkbox" class="radio" name="tag" value="housing">Housing<br>
   <input type="checkbox" class="radio" name="tag" value="other">Other<br>

   <h4>Description:</h4><textarea name="description" maxlength="500" cols="50" rows
="5"></textarea><br />
   <h4>Image:</h4><input type="file" name="image"><br />
   <input type="hidden" name="sellerID" value="<?php echo $_SESSION['user']->getID(); ?>">
   </div>
   <input class="submit_button" type="submit" name="submit">
 </form>
 </div>
 </div>
</div>
```

Controller:

Tabs are inconsistent, varies between two and four spaces.

```
public function addListing()
   {
     if(isset($_POST['submit']))
     {
```

Variable names are descriptive.

```
             $blob = file_get_contents($_FILES['image']['tmp_name']);
             $imageType = $_FILES['image']['type'];
               if(substr($imageType,0,5) == "image")
               {
```

Consistent spacing between assignment operator would improve legibility.

```
                 $res =$this->model->addListing($_POST['title'], $_POST['price'], $blob,
$_POST['description'], $_POST['tag'], $_POST['sellerID']) ;
               }
           }
           header('location: ' . URL . 'listing/addListingPage');
           if($res)
           {
             echo"your listing has been uploaded";
           }
           else{echo "we could not upload your listing";}
         }
```

Model:

Please add descriptive comments

```
public function addListing($title, $price, $image, $description, $tag, $sellerID)
   {
         Please remove old debug output
         UserID variable is no longer used, it has been replaced with user
              //print_r($_SESSION['userID']);
              $sql = "INSERT INTO listing(title, price, image, description, tag, sellerID)
VALUES(:title, :price, :image, :description, :tag, :sellerID)";
              $query = $this->db->prepare($sql);
              $parameters = array(':title' => $title,':price' => $price, ':image' => $image,
'description' => $description, ':tag' => $tag, ':sellerID' => $sellerID);
              $query->execute($parameters);
            }
```

Overall, the code is pretty consistent with the agreed style. Some additional comments would be helpful for maintenance, but the use of descriptive variables make the code easy to read.

Adherence to original Non-functional Specs

✓ = Done

✓ Application shall be developed using class provided LAMP stack

✓ Application shall be developed using pre-approved set of SW development and collaborative tools provided in the class. Any other tools or frameworks have to be explicitly approved by Marc Sosnick on a case by case basis.

✓ Application shall be hosted and deployed on Amazon Web Services as specified in the class

(Check IE) Application shall be optimized for a standard desktop/laptop browsers, and shall render correctly on the two latest versions of all major browsers: Mozilla, Safari, Chrome and IE. It shall degrade nicely for different sized windows using class approved programming technology and frameworks

✓ Data shall be stored in the database on the class server in the team's account

✓ Application shall be served from the team's account

✓ No more than 50 concurrent users shall be accessing the application at any time

✓ Privacy of users shall be protected and all privacy policies will be appropriately communicated to the users.

✓ The language used shall be English.

✓ Application shall be very easy to use and intuitive. No prior training shall be required to use the website.

· Google analytics shall be added for major site functions.

✓ Messaging between users shall be done only by class approved methods to avoid issues of security with e-mail services.

✓ Pay functionality (how to pay for good): shall be simulated this with proper UI, no backed.

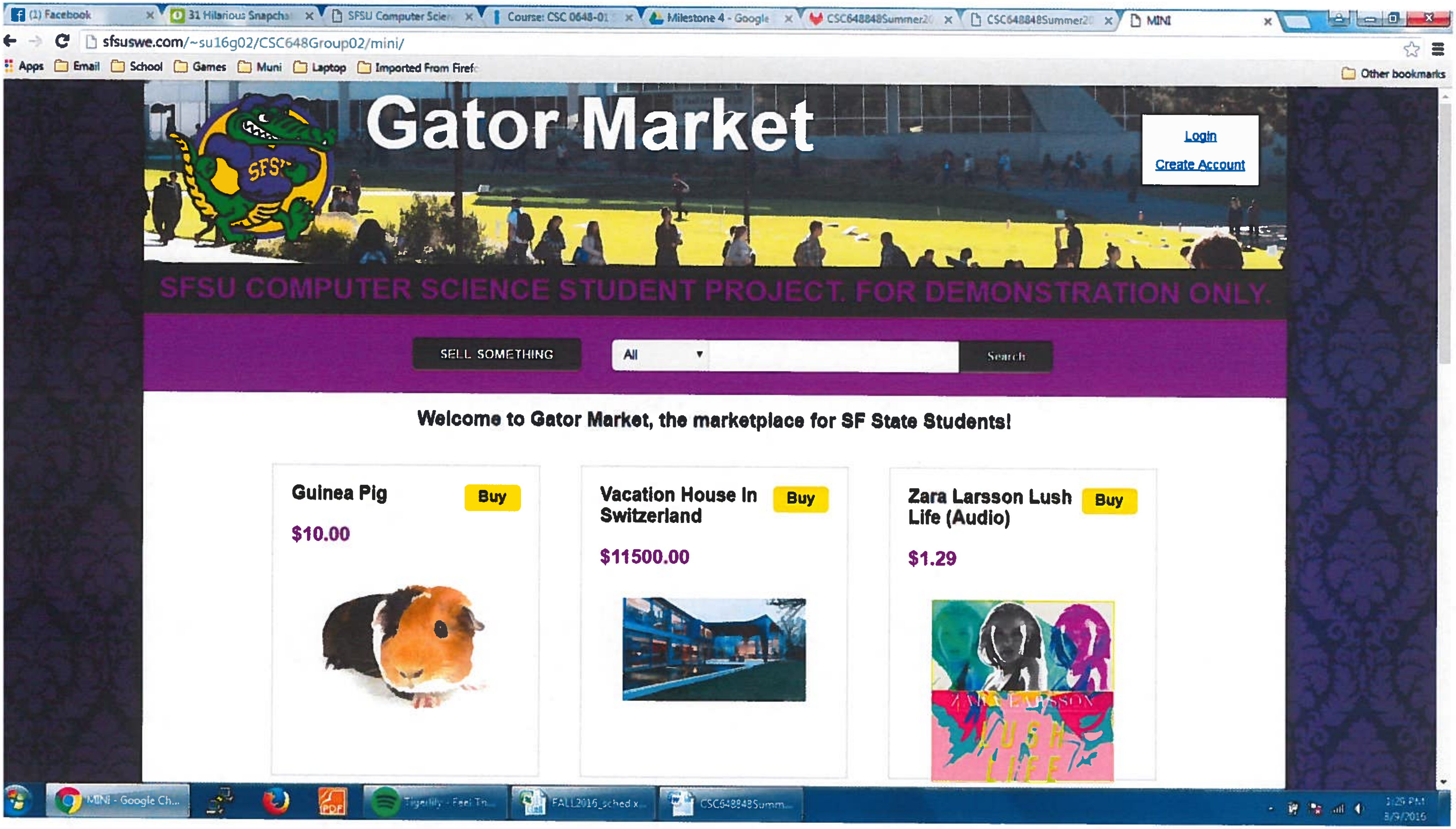
✓ Messaging between buyers and sellers: ways to do this to be specified in the class. Due to security of servers we cannot use e-mail.

✓ Site security: basic bets practices to be applied (as covered in the class)

✓ Modern SE processes and practices must be used as specified in the class, including collaborative and continuous SW development, using the tools approved by instructors

✓ The website shall prominently display the following text on all pages "SFSU/FAU/Fulda Software Engineering Project, Summer 2016. For Demonstration Only". (Important so as to not confuse this with a real application).

Product Screen Shots



Gator Market

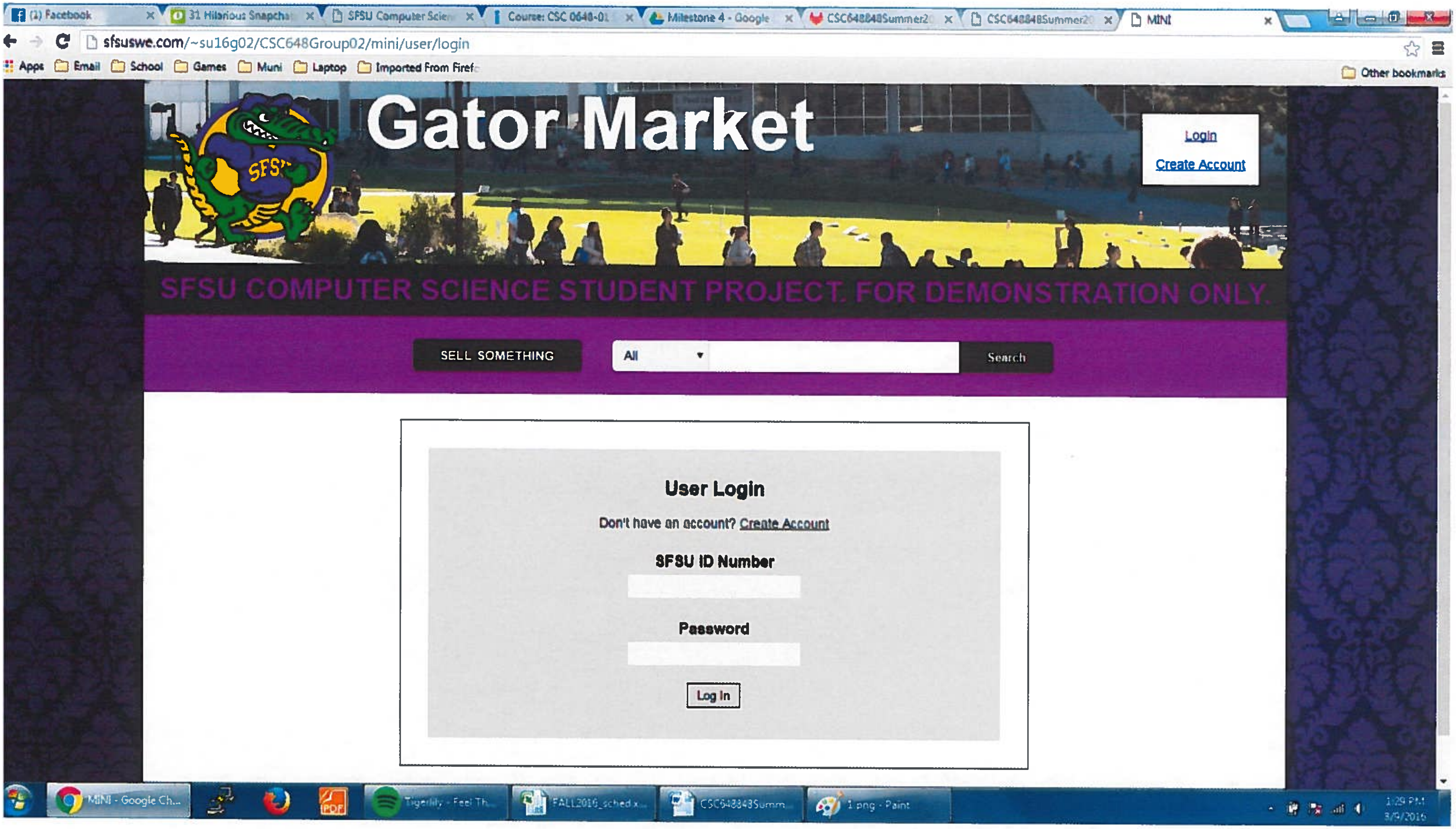
Login

Create Account

SFSU COMPUTER SCIENCE STUDENT PROJECT. FOR DEMONSTRATION ONLY.

Welcome to Gator Market, the marketplace for SF State Students!

Guinea Pig	Vacation House In Switzerland	Zara Larsson Lush Life (Audio)
Buy	Buy	Buy
$10.00	$11500.00	$1.29



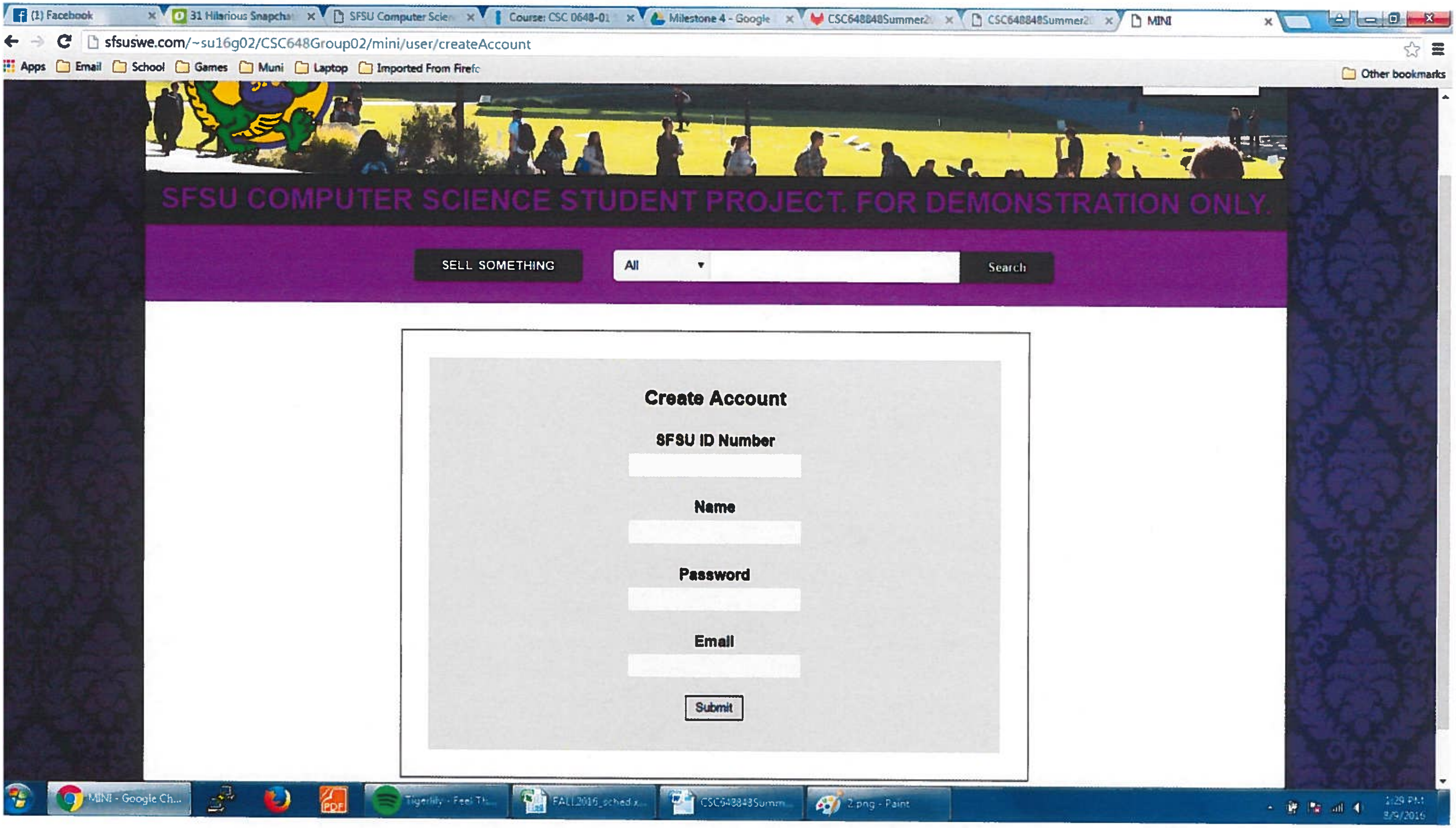
User Login

Don't have an account? Create Account

SFSU ID Number

Password

Log In



sfsuswe.com/~su16g02/CSC648Group02/mini/user/createAccount

SFSU COMPUTER SCIENCE STUDENT PROJECT. FOR DEMONSTRATION ONLY.

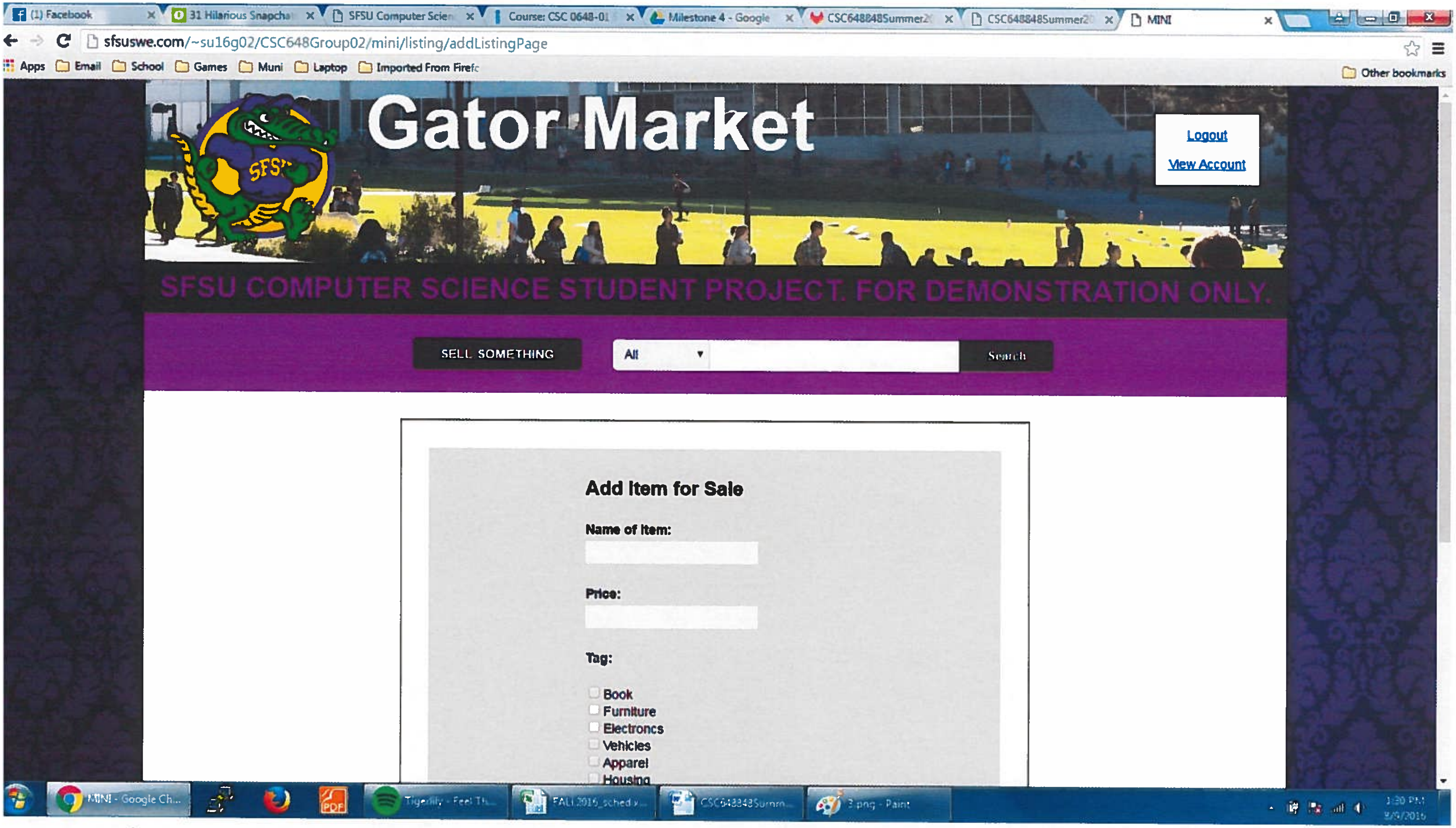
Create Account

SFSU ID Number

Name

Password

Email



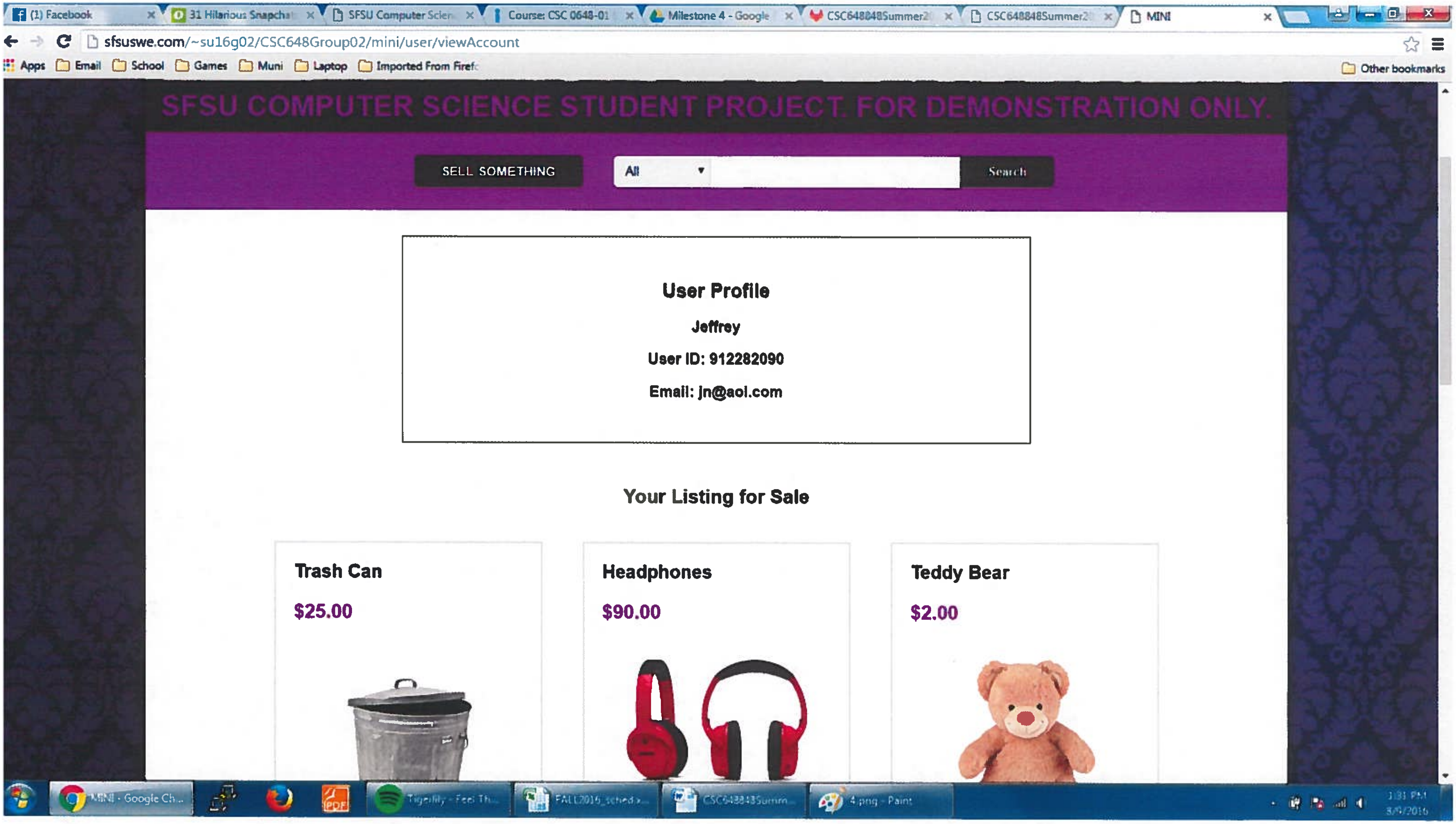
Add Item for Sale

Name of Item:

Price:

Tag:

□ Book
□ Furniture
□ Electroncs
□ Vehicles
□ Apparel
□ Housing



SFSU COMPUTER SCIENCE STUDENT PROJECT. FOR DEMONSTRATION ONLY.

User Profile

Jeffrey

User ID: 912282090

Email: jn@aol.com

Your Listing for Sale

Trash Can

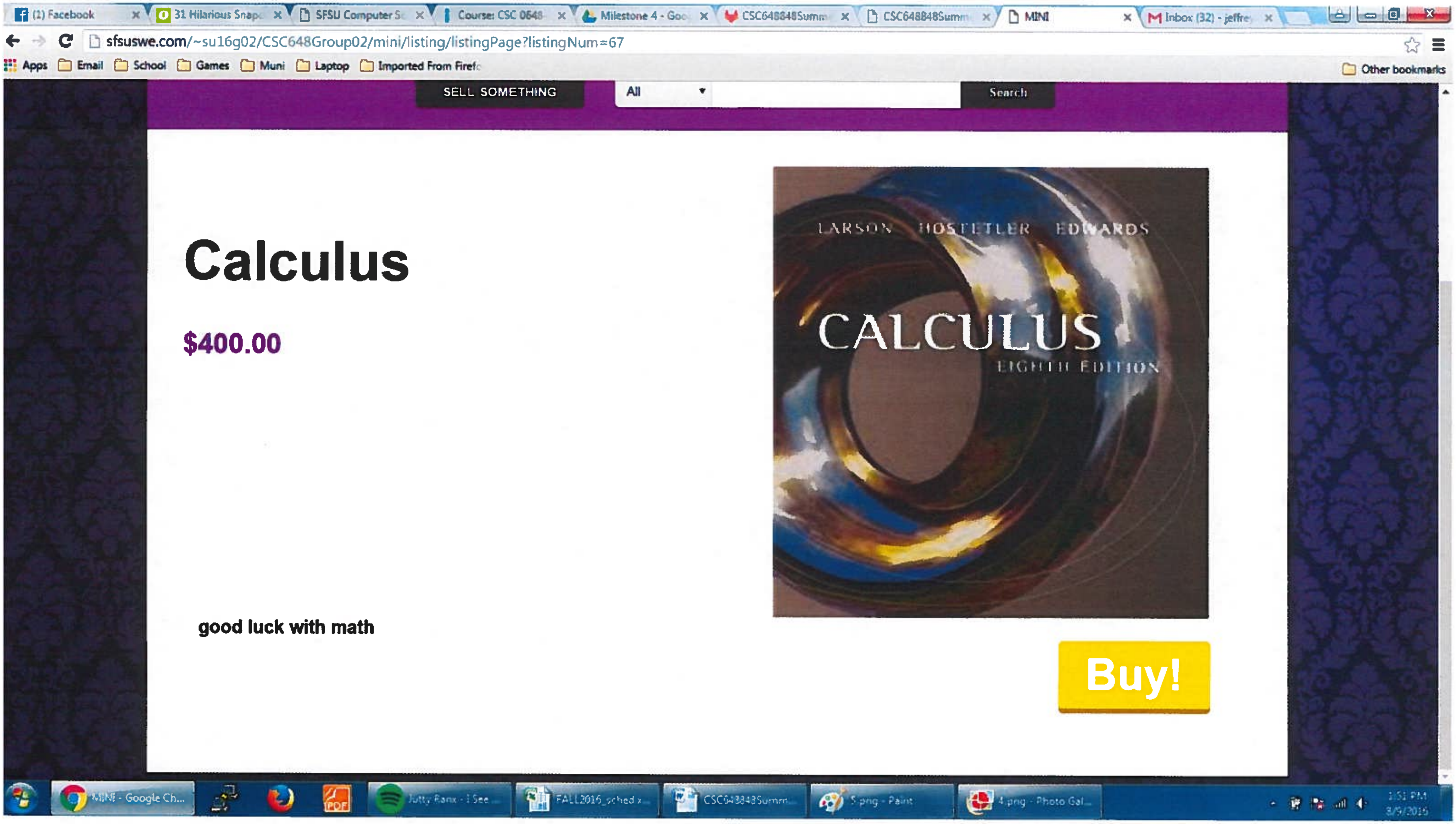
$25.00

Headphones

$90.00

Teddy Bear

$2.00

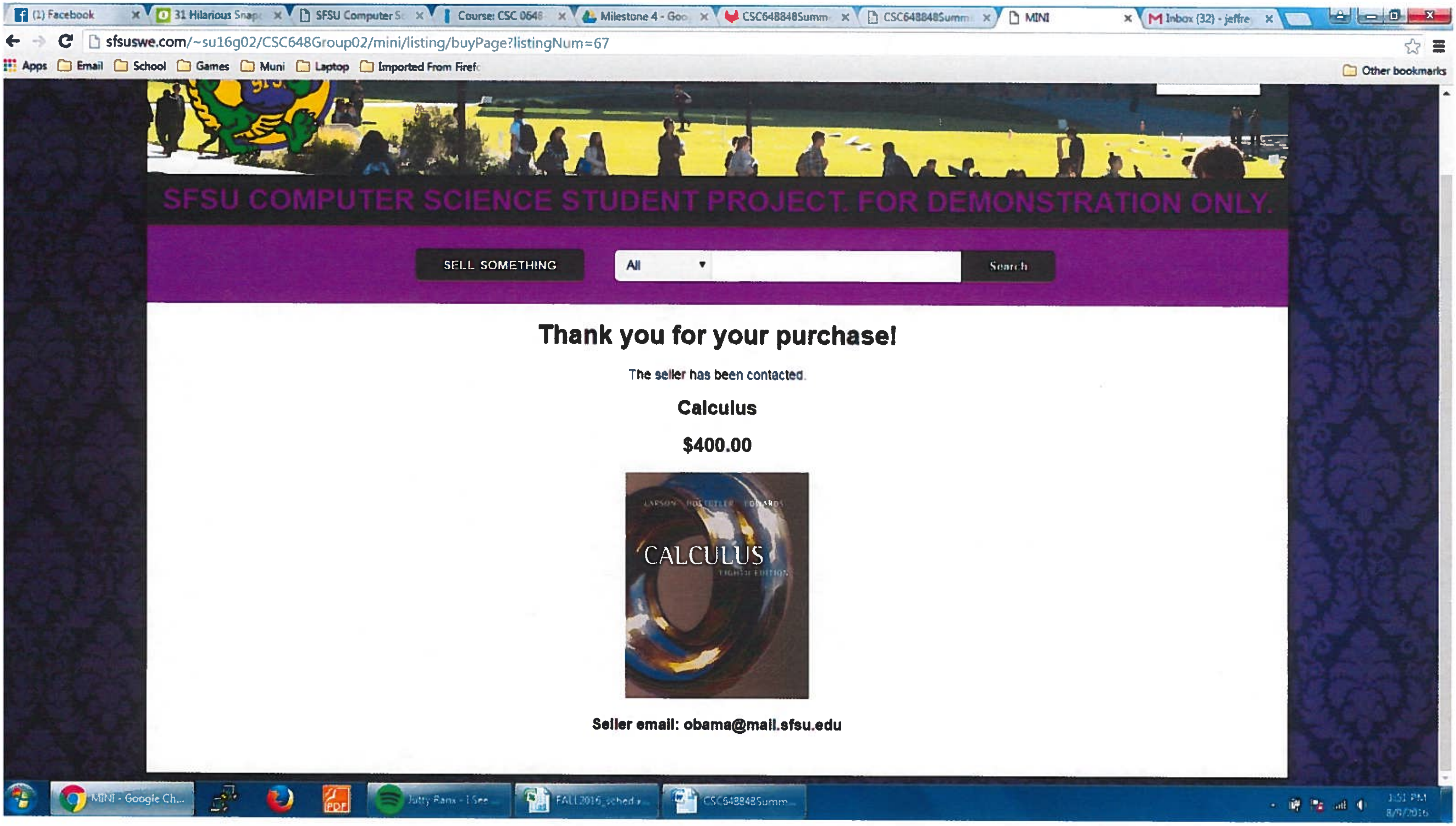


sfsuswe.com/~su16g02/CSC648Group02/mini/listing/listingPage?listingNum=67

Calculus

$400.00

good luck with math



Thank you for your purchase!

The seller has been contacted.

Calculus

$400.00

Seller email: obama@mail.sfsu.edu

Team Member Contribution

Gmail **jeffrey noehren <jeffreynoehren9@gmail.com>**

Project Contributions for M5 Document

6 messages

jeffrey noehren <jeffreynoehren9@gmail.com> Tue, Aug 9, 2016 at 1:44 PM
To: Jason Newman <gojirra@gmail.com>, Olatope Oladipupo <olatopetyler@gmail.com>, Maria Lienkaemper <mlienkae@mail.sfsu.edu>, Kenny Tran <kethtran@gmail.com>

Hey everyone,

Thank you guys for being such an awesome team this semester! For our final M5 document we need to have an Individual Contribution Page where everyone submits in this email chain to everyone in the group a brief summary of what they contributed to the project. You are more than welcome to do this in bullet point form if you please.

Because our document is due tomorrow please do this ASAP so I can bind our final document later today!

Thanks,
Jeffrey

jeffrey noehren <jeffreynoehren9@gmail.com> Tue, Aug 9, 2016 at 1:48 PM
To: Jason Newman <gojirra@gmail.com>, Olatope Oladipupo <olatopetyler@gmail.com>, Maria Lienkaemper <mlienkae@mail.sfsu.edu>, Kenny Tran <kethtran@gmail.com>

Jeffrey's Contribution:

- Set up the MINI framework on our server
- Connected our SQL Database to Workbench
- Created the ListingObject Class in the Model
- Created the listing.php controller file and all functions within
- Built the Functionality for the Add Listing Page, Browse, and Search Features
- Git Master

[Quoted text hidden]

Jason Newman <gojirra@gmail.com> Tue, Aug 9, 2016 at 5:17 PM
To: jeffrey noehren <jeffreynoehren9@gmail.com>
Cc: Olatope Oladipupo <olatopetyler@gmail.com>, Maria Lienkaemper <mlienkae@mail.sfsu.edu>, Kenny Tran <kethtran@gmail.com>

Contributed to M1 and M2 documents.
Site mock ups.
Overall site look and feel.
Implemented much of the CSS and front end elements.

--
Jason Newman
Designer
+1.925.878.9629
jnewmandesign.com
gojirra@gmail.com
[Quoted text hidden]

Olatope Oladipupo <olatopetyler@gmail.com> Tue, Aug 9, 2016 at 5:51 PM
To: Jason Newman <gojirra@gmail.com>
Cc: jeffrey noehren <jeffreynoehren9@gmail.com>, Maria Lienkaemper <mlienkae@mail.sfsu.edu>, Kenny Tran <kethtran@gmail.com>

Milestone 1,2,4 Documents
Single listing page: php and css
[Quoted text hidden]

Maria Lienkaemper <mlienkae@mail.sfsu.edu> Tue, Aug 9, 2016 at 6:06 PM
To: Jason Newman <gojirra@gmail.com>, jeffrey noehren <jeffreynoehren9@gmail.com>
Cc: Olatope Oladipupo <olatopetyler@gmail.com>, Maria Lienkaemper <mlienkae@mail.sfsu.edu>, Kenny Tran <kethtran@gmail.com>

- Implemented the UserObject in the model
- Implemented the user controller
- Implemented create account functions and page
- Created buy page functions
- Created view account functions
- Security fixes; has password, limit input in text fields
- Search fix, page refresh fix, and other bug fixes

[Quoted text hidden]

Kenny Tran <kethtran@gmail.com> Tue, Aug 9, 2016 at 6:24 PM
To: Jason Newman <gojirra@gmail.com>
Cc: jeffrey noehren <jeffreynoehren9@gmail.com>, Olatope Oladipupo <olatopetyler@gmail.com>, Maria Lienkaemper <mlienkae@mail.sfsu.edu>

Contributed to M1-M4 documents.
Wrote documentation for code.
Helped populate database with listings.

[Quoted text hidden]