SW Engineering CSC 648/848 Section 02

Spring 2017

Gator Sales

A one-stop e-shop where San Francisco State University
Students can buy and sell used and new items.

Milestone 2

March, 16 2017

Team 09

Charles Williams (aasim@mail.sfsu.edu)
Jason Huang
Myat Min Maung
Krishnan Ramakrishnan
Leopoldo Rodriguez
Grant Gaviglio
Wilson Le

Revision history

Date	Description
February 28, 2017	M1 Rough draft for in-class feedback
February 29, 2017	M1 Final draft submission
March, 14 2017	M2 Rough draft for in-class feedback
March, 16 2017	M2 Final draft submission

Executive summary

Gator Sells is an e-shop that allows students to buy or sell products to each other within the same campus. This e-shop allows students to post, view and buy items without difficulty. While other college campus e-shop sites such as Student Listings and UOW Students Buy and Sell exist, Gator Sells features a filter system that allows a user to quickly browse through items that a student is interested in. Gator Sells also features a private messaging system that allows a buyer and seller to quickly and reliably communicate. While Gator Sells does not have all the features that bigger e-shops such as Amazon or E-bay has, Gator Sells is currently specifically catered towards students currently attending San Francisco State University. By requiring students to register with a San Francisco State University e-mail address, the system can ensure that users are from the university, as well as a way to verify a user's identity. Furthermore, it is possible to expand the use of Gator Sells in the future for use in other campuses by changing the e-mail filter when registering for an account with Gator Sells.

Gator Sells is currently being developed by a seven student start-up team in San Francisco State University. The students have a deep understanding about San Francisco State University, and can tailor the site to be attractive towards San Francisco State University. The students can also advertise the website to their peers, which may increase the user base of the e-shop for the campus. Any feedback that the group receives is received directly from the students that are using the site, allowing the student group to quickly and efficiently fix and change the items in the e-shop as needed.

User Cases V2

Buyer:

1. **Unregistered user**: Maria and her roommate Jill decided they needed a new TV for the living room. One day Maria finds a flyer on campus about a new website exclusive to students where they can **sell** their **new** and **used items** to other students, upon getting home Maria informed her roommate about the website which they are both interested in searching for potential TV's. They started out using the **search** to narrow down the **items** listed to just the desired TV, and using the **filters** to find the right size and color. After the girls found the TV they wanted they decided to **message** the **seller** for more information, but upon doing so the girl were prompted to **register**. After registering the girls were able to message the **seller**.

2. **Registered user:** Jen is an international student who just moved to San Francisco. She goes onto the Gator Sells website to find available living room sets. After scrolling through some options using the available **filters** (price, , color), she finds a few bed room sets that she likes and messaged **seller** to express her interest in the set, which will **send** an **message** through the website to the **seller**.

Seller:

3. Kimberly is a senior in her last term at SFSU and wants to **sell** her **items** before moving back home. She likes the security of website whereas only student with a valid @mail.sfsu.edu can **buy** or **sell items** on the site. Upon clicking the sell button she is prompted to register with her student email and password, after registering with the site she is able to create a new posting by upload her **items** listing, along with pictures and contact information.

Administration:

4. Bob, an administrator of Gator Sells, has received a report from a user that a post to sell an item was fraudulent and the information on the post is inaccurate. Bob can log into his administrator account in Gator Sells and go to the seller's post that had been reported. Bob can then look at the information that was posted by the seller and see if there was anything suspicious on the post. If Bob determines that the information on the post is inaccurate, Bob can delete the post and send a warning to the seller. If the seller has received several warnings beforehand, Bob can then choose to ban the seller from the site itself.

Data Definition V2

Important terms:

1. **Gator Sells**: The name of our e-store.

2. **Registration**: Process in which the user creates a User Profile. The User Profile will consist of First name, Last name, Email, and Password, Photo for the User Profile will be optional.

3. **Items**: Type of items that are posted on the website. Which can be Books, TVs, Computers, Cell phones, Furniture, or Video Games 'items' will be searchable through attributes such as:

    1. Price
    2. Items (Electronics, Books, Room)
    3. Location (Address, Street, City, State, Zip code)
    4. Electronics (TVs , Computers, Cell phones, Video Games)
    5. Clothing (Male/Female/Shirts/Pants/ Accessories)
    6. Books (Used/New)
    7. Computers (Desktop, Laptop)
    8. Cell phones (Android, IPhone)
    9. Furniture (Beds, Mattress, Table, Couch)
    10. Video Games(Used/New)
    11. Distance from campus (Default, can be changed)
    12. Date item posted

4. **Media**: Photo of items uploaded by seller.

5. **Shopping Cart:** All the buyer's items will be temporarily stored here before the checkout process.

6. **One-Click Checkout:** This process will short circuit the conventional purchasing process, the buyer may go straight to paying process and complete the procedure in a shorter amount of time. [buyer shall only purchase one product at a time with this process flow]

Users:

Types of users in relation to the project include:

- **Seller**: **User** who **posts** one or more **items** on the website.
- **Buyer**: **User** that **buy** posted **items** from the **sellers**.
- **Admin**: **User** with administrative privileges that Moderates the website and is able to remove false postings.
- **Unregistered user**: Users who just want to browse the website. They are not required to go through **registration**.

Database Schema

Database tables and associated column names

1. **User**

first_name	First name of the user
last_name	Last name of the user
User_id (PK)	Unique identifier of an user (PK)
Student_id (UI)	Unique student identifier provided by SFSU
email	Email address of user
password	User's password
user_image	Photo image of user.

2. **Item**

Item_id (PK)	Unique identifier of item
name_of_item	Short name of an item
location	Name of the location where item is.
price	Unit price of item
description	Description of item.
item_photo	Photo image of item.
Category_ID (FK)	Unique identifier of item category (Lookup table)
used_or_new	Identifies if item is new or used.
Item_list_date	Date on which item was posted.
Item_Sold_Date	Date on which item was sold
Distance_From_Campus	Distance of item's location from SFSU.

3. Category

category_id	Unique identifier of an item category or subcategory.
category_description	Description of item category
Parent_category_id (FK on category_id)	Category identifier of parent category.

4. Shopping Cart

Cart_id (PK)	Unique identifier of user's shopping cart
User_ID (FK)	User identifier (Lookup table)
Date_Created	Date on which shopping cart was created.
shopping_cart_expiration_date	Date on which shopping cart is expired
Shopping_cart_total	Total amount of items in the cart.
Number_of_items	Number of items in shopping cart.

5. Shopping cart details

Shopping_Cart_Detail_ID (PK)	Unique identifier of shopping cart details
Shopping_Cart_ID (FK)	Shopping cart identifier (Lookup table)
Item_identifier (FK)	Item identifier (Lookup table)
Item_amount	Cost of item being bought

Functional Specifications V2

Buyer:

1. **Browsing:** The buyer shall browse our website's to view item listing without the need to register for an account.

2. **Registering and/or logging in:** The buyer shall use an SFSU email address to create an account to access the site the buy and sell feathers, or to contact a seller. Users shall have the ability to log in to or log out from the website from any page on the website.

3. **Sorting:** The buyer shall be able to filter postings by clicking buttons related to the search metadata listed in the data definition.

4. **Private messaging:** The buyer shell have the ability to contact potential buyers and sellers.

5. **Item Location:** The users shall see the item's distance from SFSU or any desired set location.

Seller:

1. **Post Items:** The seller shall login to post items on site in which they want to sell.

2. **Media posting:** The seller shell have the ability to post photo of items uploaded.

3. **Private messaging:** The seller shell have the ability to reply to potential buyers and.

Administration:

1. **Administrative privileges**: The admin shall have administrative privileges in which to moderates the website and shall be able to remove inappropriate postings; another to remove inappropriate user or that don't follow the sites Terms and Condition.

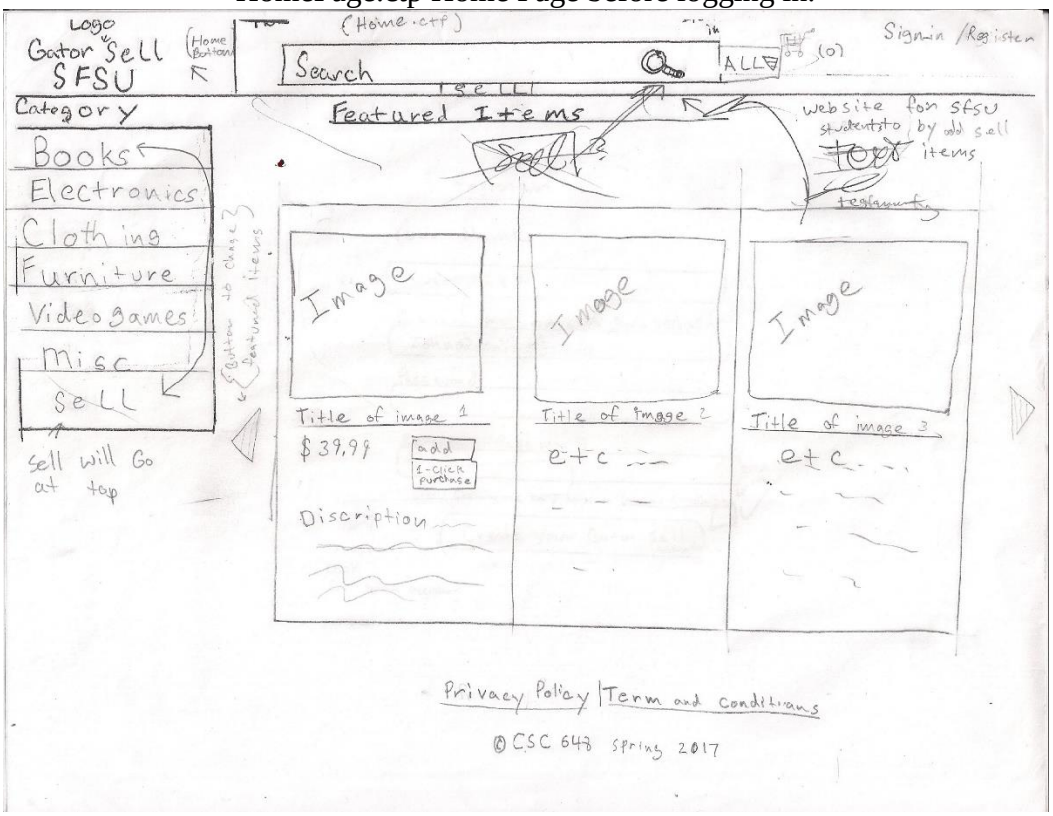
Priority 1:	
Buyer:	
Unregistered user:	
	Browse site to view item listing without the need to register for an account.
	Use the search bar to look for items.
	Search for items by category.
	Create an account (if also SFSU student).
Registered user:	
	All of the above plus
	Access his/her shopping cart
	Buy items
	Edit the shopping cart

	Message seller
Seller:	
Registered user:	
	All of the above plus
	Post items on site in which they want to sell.
	Post photo of items uploaded.
Administrator:	
	All of the above plus
	Manage the database directly
	Remove inappropriate postings
	Remove inappropriate user
Priority 2:	
Unregistered user:	
	Sort items by price and date
Registered user:	
	All of the above plus
	Rate available items
Priority 3:	
Unregistered user:	
	Sort items by rating
Registered user:	
	All of the above plus
	Flag posts as inappropriate
Administrator:	
	Suspend users

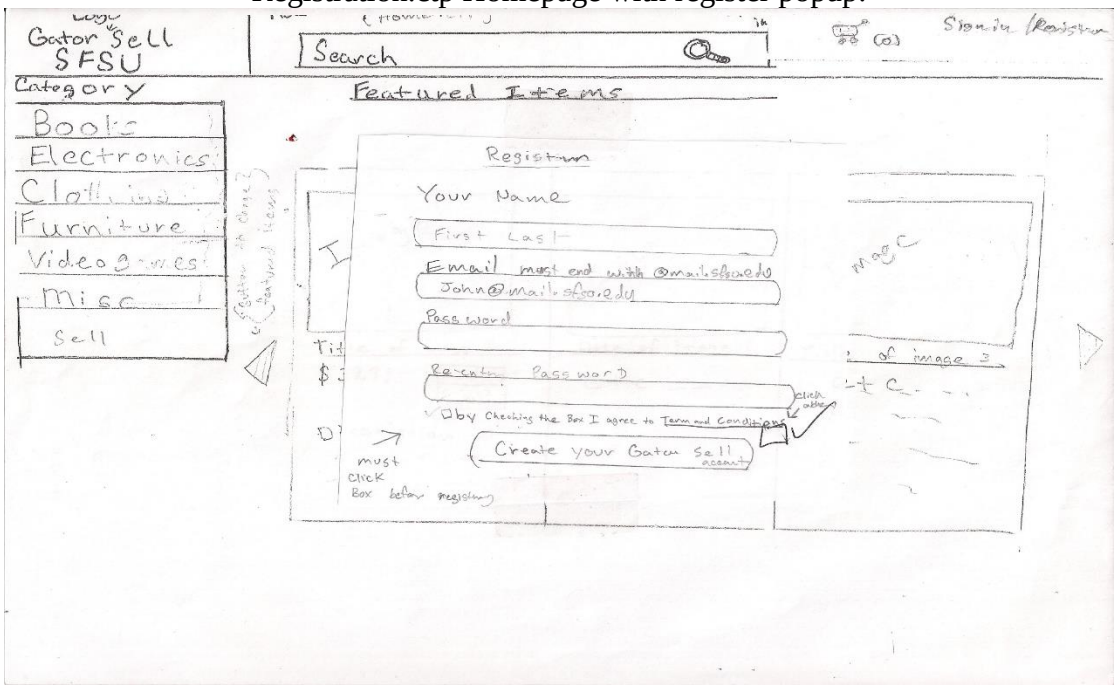
Non-functional Specifications V2

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, and 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).

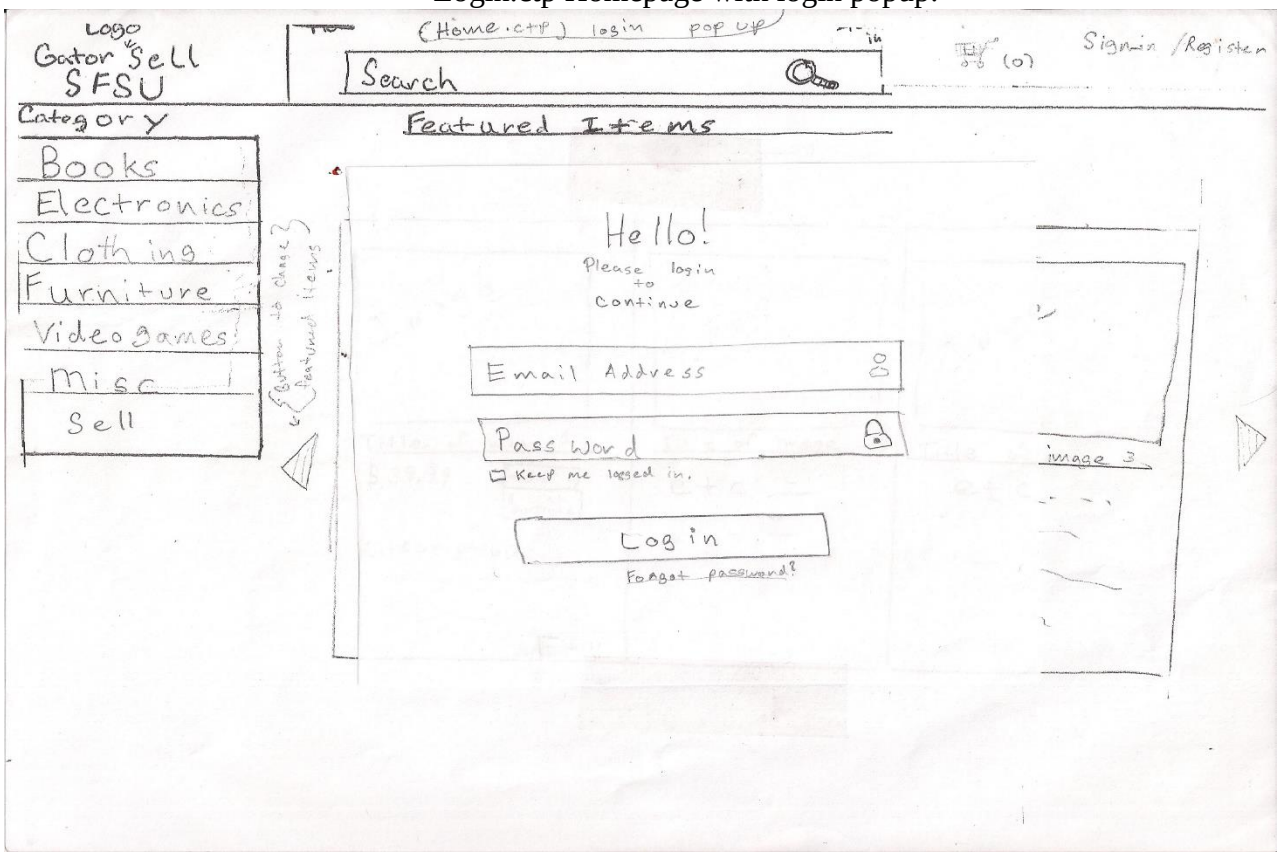
HomePage.ctp Home Page before logging in.



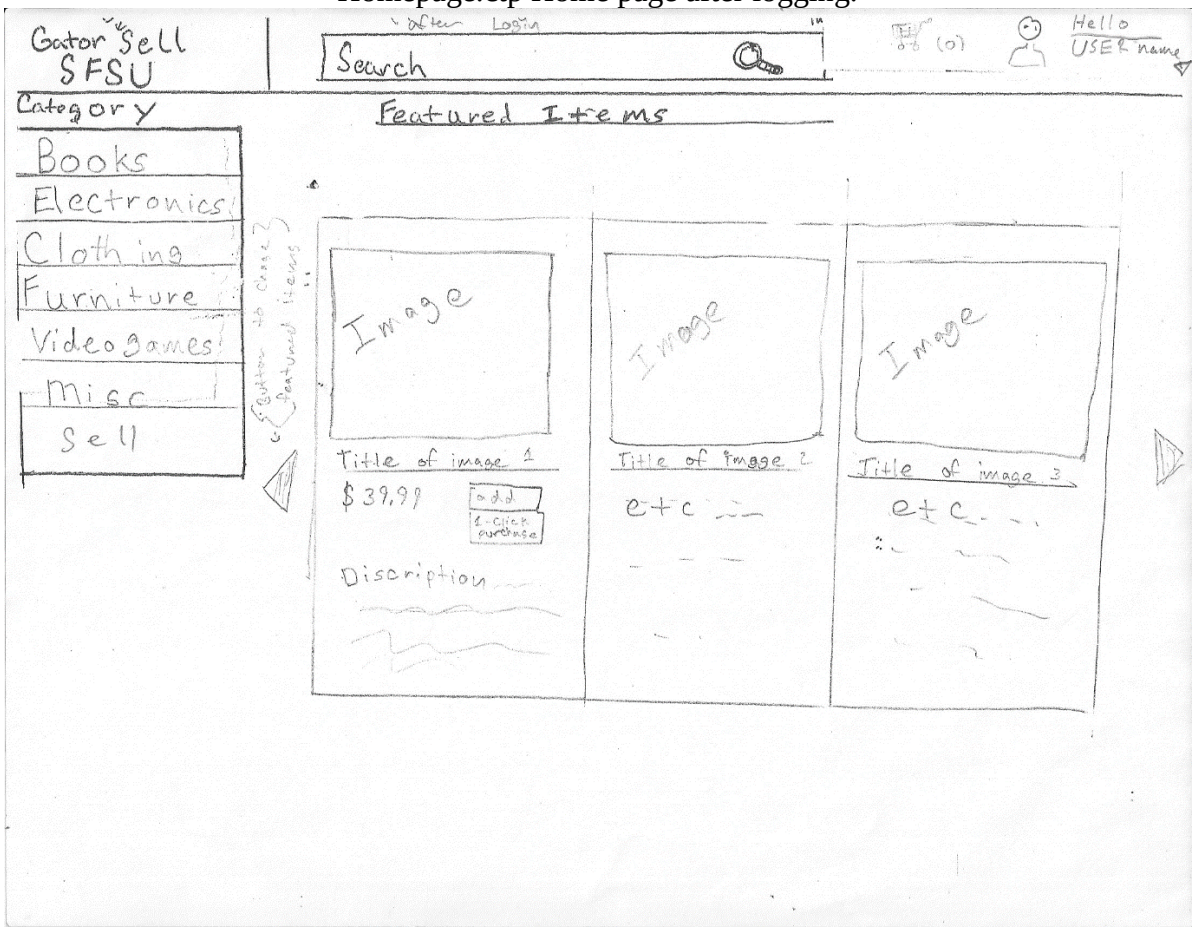
Registration.ctp Homepage with register popup.



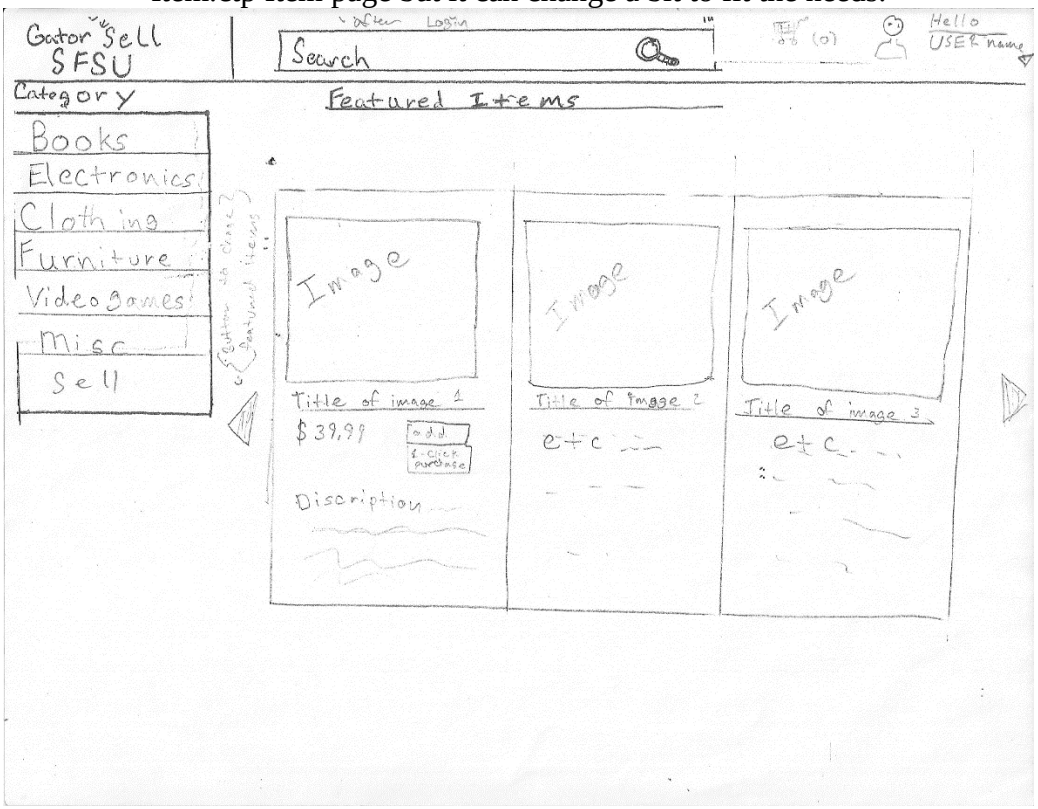
Login.ctp Homepage with login popup.



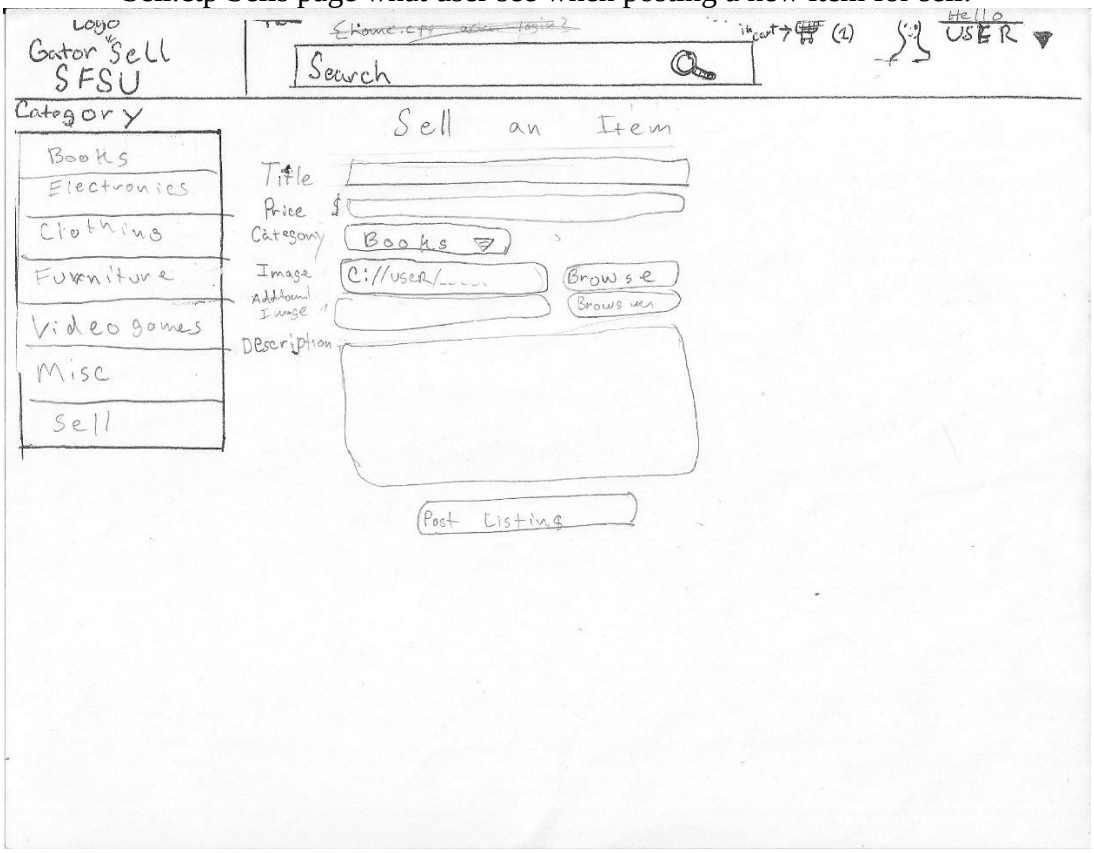
Homepage.ctp Home page after logging.



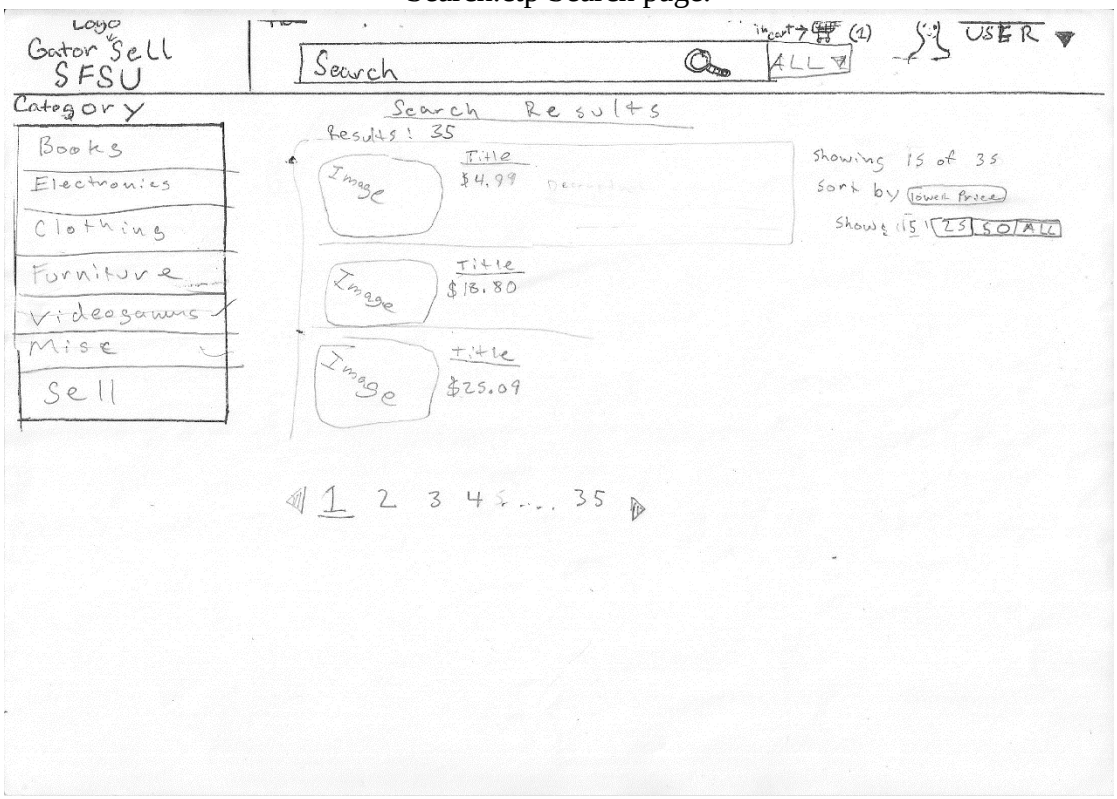
Item.ctp Item page but it can change a bit to fit the needs.



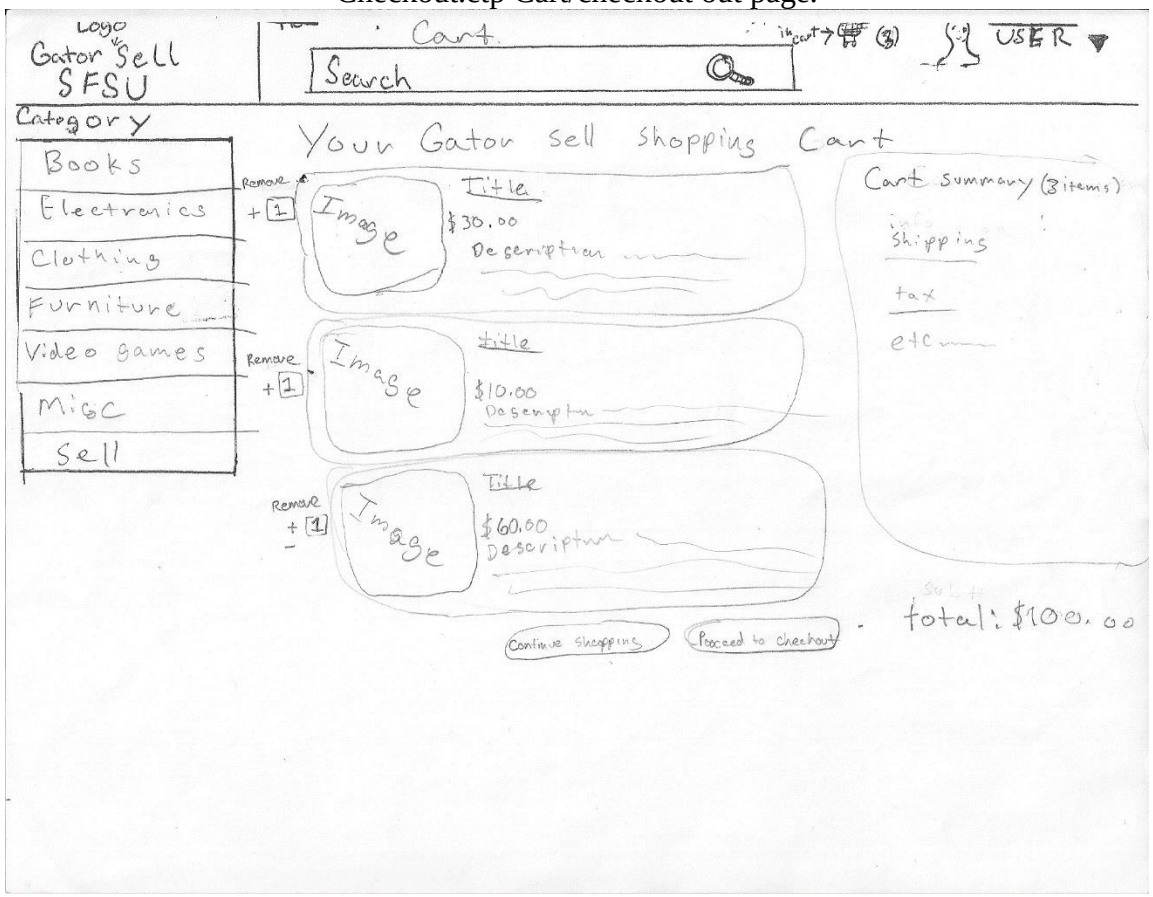
Sell.ctp Sells page what user see when posting a new item for sell.



Search.ctp Search page.

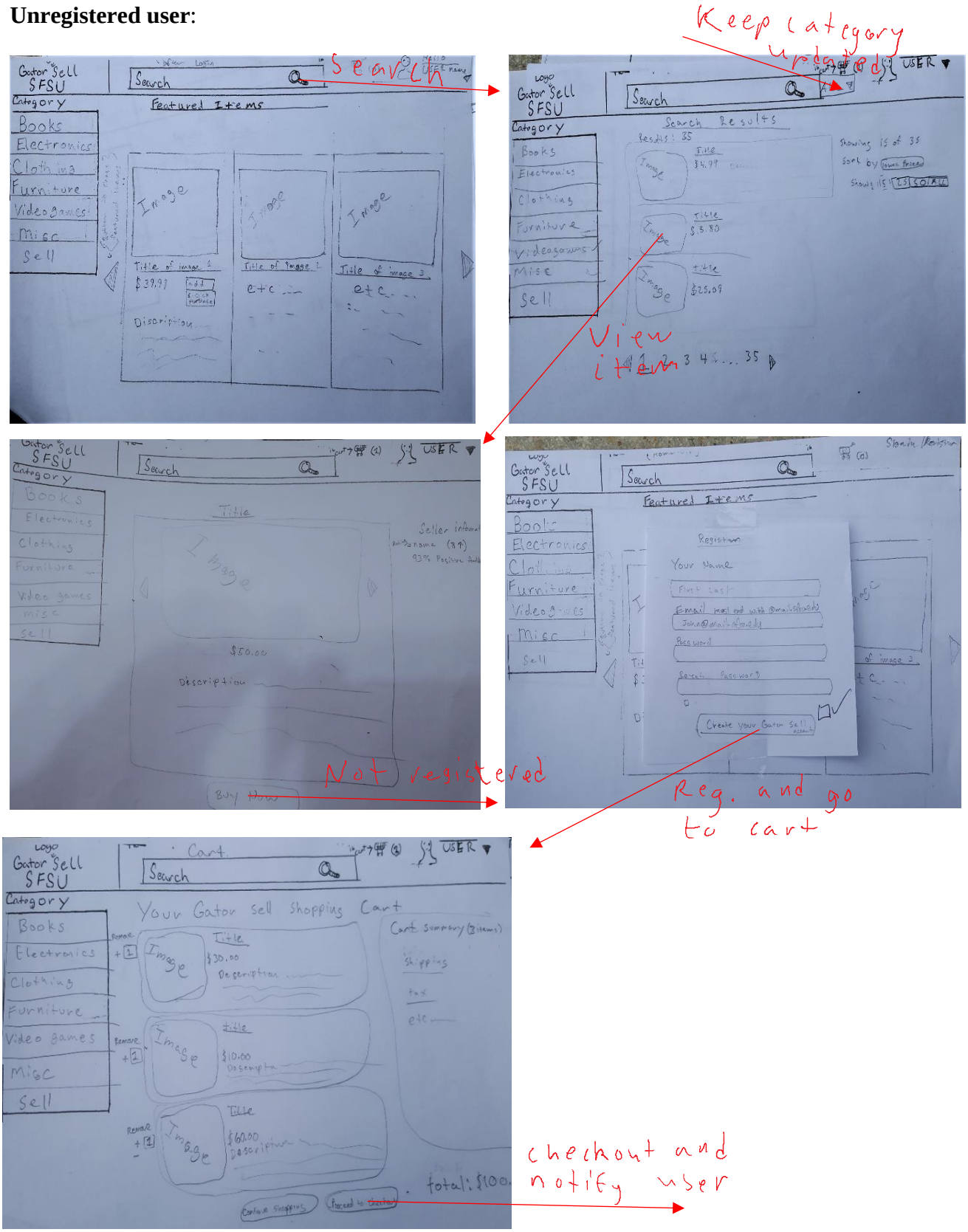


Checkout.ctp Cart/checkout out page.



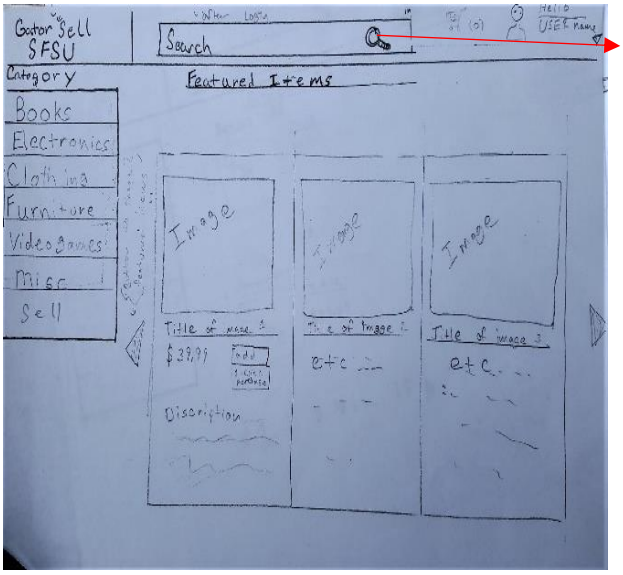
Buyer:

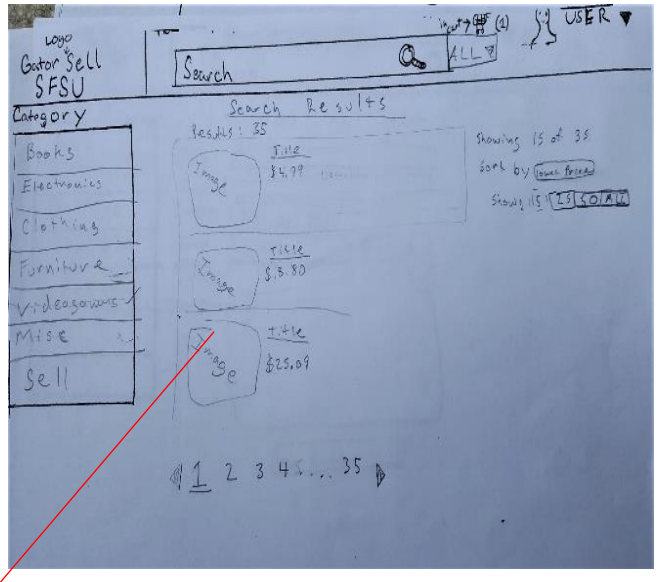
Unregistered user:

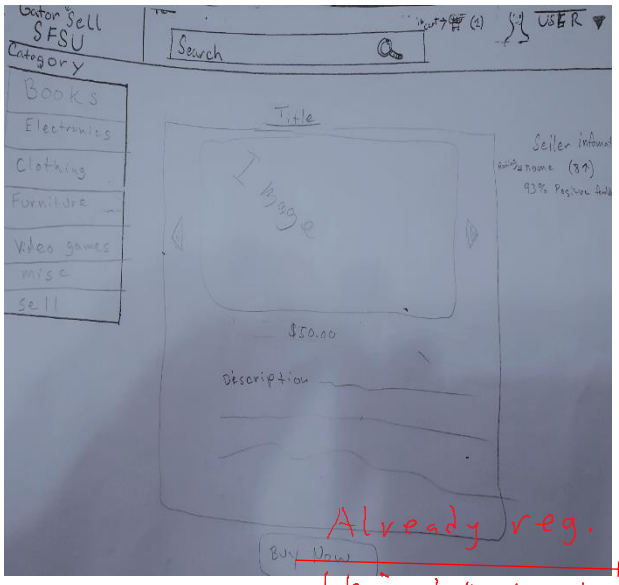


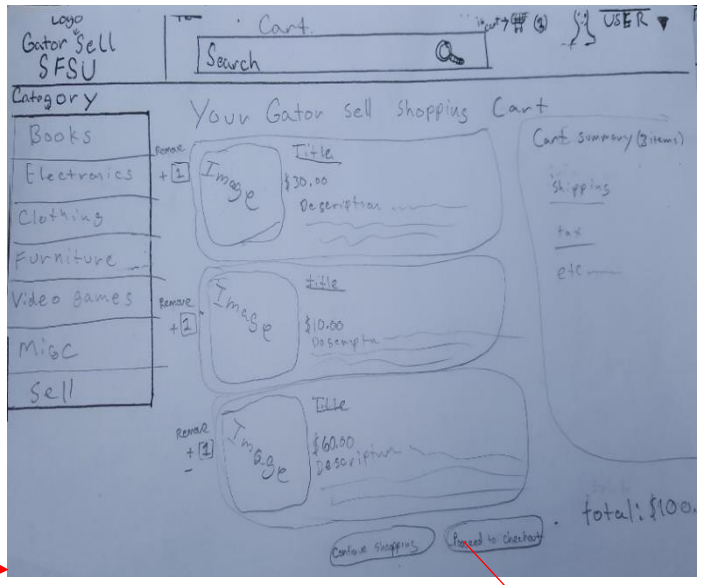
Buyer

Registered user:



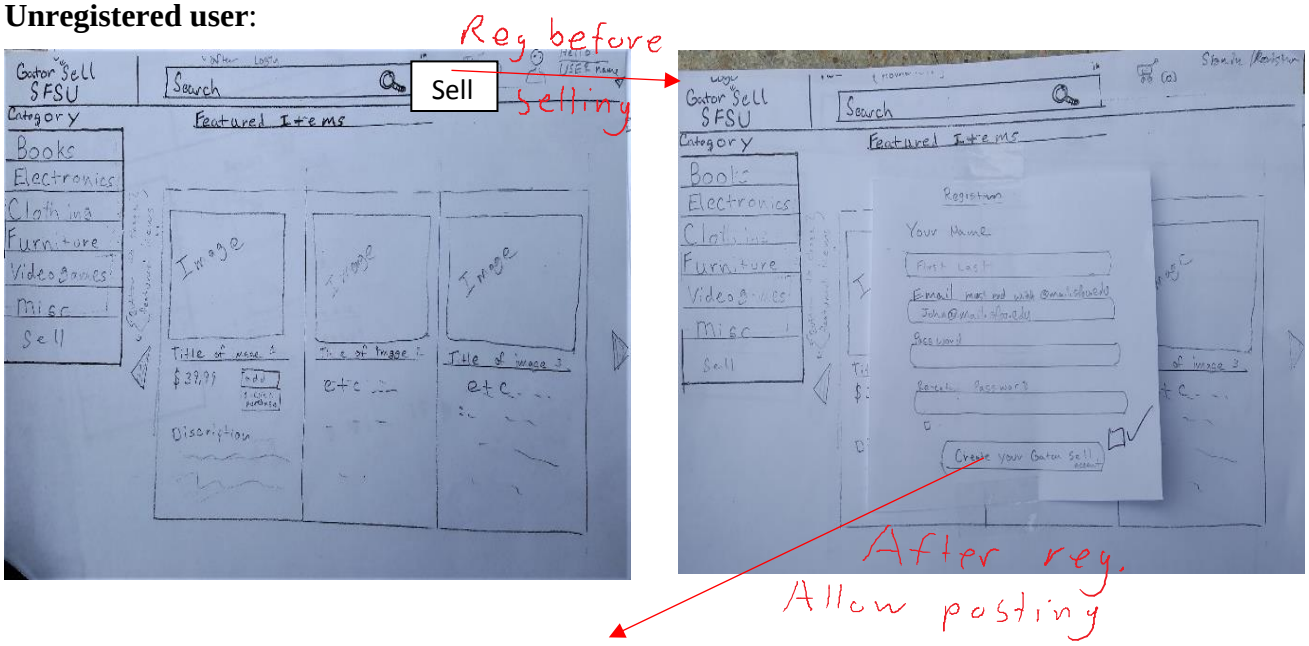


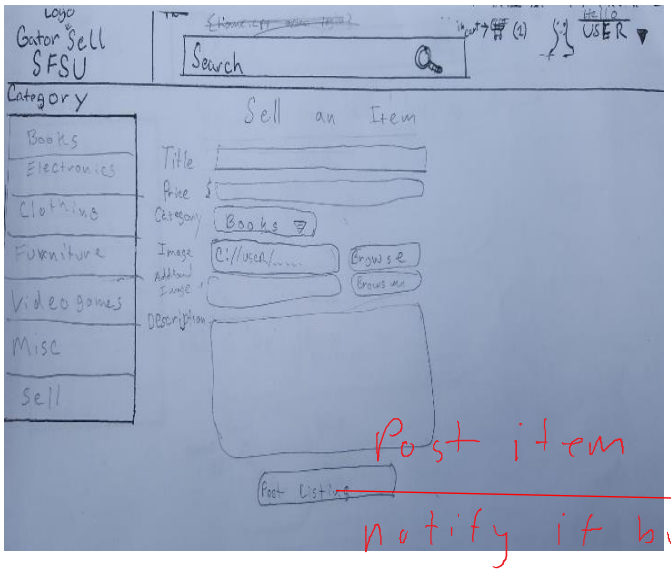




Seller:

Unregistered user:





Competitive Analysis

Name of Site	Filters	Search	Price Comparison	Communication With seller	*One-Click Checkout	*Domain Specific Email Registration
Gator Sells	✓✓	✗	✓	✓	✓	✓✓
Student listings	✗	✓	✓	✗	✗	✗
UOW Students Buy and Sell	✗	✓	✗	✓	✗	✗
The Book Exchange Network	✗	✓	✓	✗	✗	✗
Amazon	✓✓	✓✓	✓	✓	✓	✗
Ebay	✓	✓✓	✓	✓✓	✗	✗
Sell Student Stuff	✓	✓	✓	✓	✗	✗

✓: Feature included ✓✓: Feature included excel ✗ : Feature not included *: Special Features

Gator Sells is an e-shop that is used exclusively by San Francisco State University students looking to buy or sell items used and new. Gator Sells features a registration system that requires users to use a @mail.sfsu.edu email address to register. This system is in place to ensure that all posts and bids by buyers and sellers are by San Francisco State University students or staff. With the option to buy an item with one-click makes the process of buy much faster for the user. Gator Sells features a filter system that allows users to quickly and efficiently search for an item in a filtered category. This filter system gives Gator Sells an edge above the current competition such as Student Listings, University of Wollongong Students Buy and Sell, and The Book Exchange Network. While Gator Sells is not as efficient as bigger e-shops such as Ebay or Amazon, Gator Sells is catered specifically towards the needs and wants of San Francisco State University students. The team that is developing Gator Sells is able to receive feedback directly from its users, and is able to quickly and effectively cater to the needs of its user base.

High-level system architecture

Development technologies:

- The application shall be developed using the LAMP stack as suggested by the stakeholders, which includes: Linux, Apache, MySQL and PHP.

- The main framework that shall be used in the development of this application is the PHP CakePhp framework.

- Technologies that shall be used in the development of this application are JQuery, Bootstrap and HTML/CSS.

Third party APIs:

- The application shall make use of certain APIs which include: Google Maps, and Google Plus.

Browser support:

- This application shall support the current and one older versions of popular browsers at the time of testing – namely Chrome, Firefox and Safari.

Admin interface:

- The admin interface for the application shall be managed through MySQL Workbench.

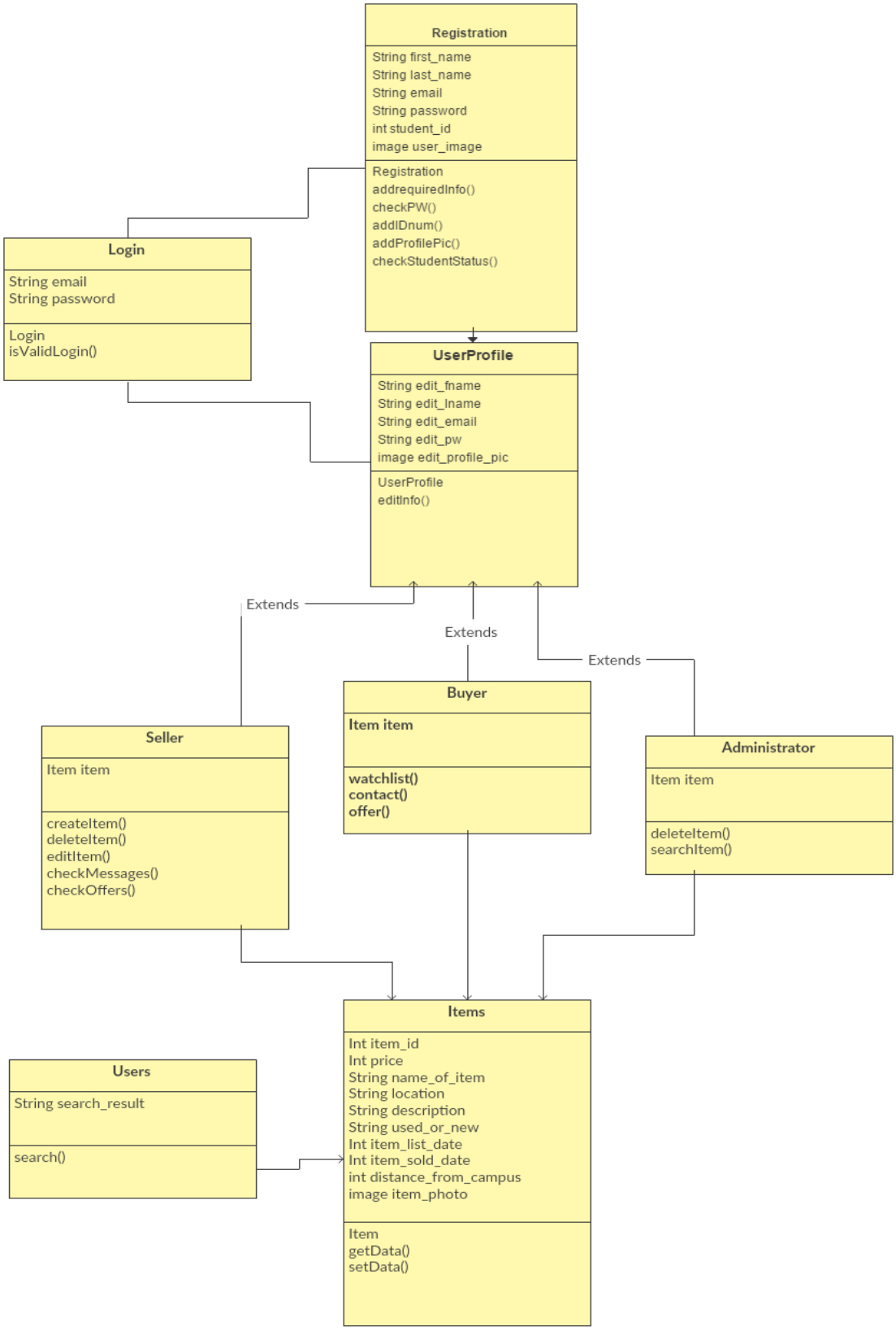
Tools and IDEs:

- The tools and systems that shall be used towards the production of this system are the terminal/command line. Code editors or IDEs are based on the developer's discretion.

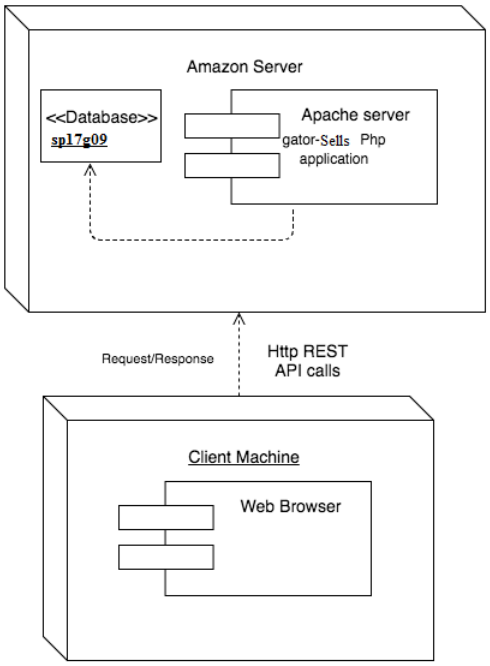
Version control:

- The code for this application shall be stored on our private project (repository) on Gitlab.
- Team members shall collaborate together on Gitlab.

High Level UML Diagrams



UML Component and Deployment Diagram



High-level APIs

Item

1. addItem(int item_id, float price, string description, string location, string used_or_new, int distance_from_campus, string item_address, int item_list_date, string item_photo, int category_ID)
    - Accepts a set of parameters describing an item and creates a new record of the item.

2. getAllItems()
    - Returns a list of all the items sorted in the specified order by the specified column. Returns all items if no criteria is given.

3. getItems(int max, string sortColumn, int sortOrder)
    - Returns a list of maximum 'max' items sorted by sortColumn in increasing or decreasing order specified by sortOrder.

4. getItems(int id)
    - Returns a single item identified by id.

5. getItemsBySelf()
    - Returns a list of items created by the requesting user.

6. getItemsFromSearch(String query)
    - Returns a list of items if values in one or more columns of interest match the words given in a query string.

7. getFavoriteItems()
    - Returns a list of items marked as 'Favorite' by the requesting user.

User

1. addUser(String first_name, String last_name, String passwordHash, int User_id, email, int Student_id)
    - Adds (Registers) a new user in the database, given a unique username.

2. getUser(String email, String password)
    - Returns a user record if the username matches the password.

Media

1. addMedia(String path, int type)
    - Adds (Creates) a new media record into the database specified by the relative path and type.

2. getMediaByListingId(int listingId)
    - Returns a list of media records associated with the given item ID.

3. getMedia(int id)
    - Returns a media entry record identified with the given media ID.

Category

1. addCategory(String category_id, String category_description, int parent_category_id)
    - Adds (Creates) a new category entry with the given category_description, a category_id will be generated by sequential order.

2. getCreatedCategories()
    - Returns a list of category entries created by the user (logged in).

Administrator

1. deleteCustomer(customer_id Int)
    - Return: fail or success

2. editCustomer(email String, password String, phone String, street String, city String, state String,
    - zipCode String)
        Return: 0 (fail) or 1 (success)

3. deleteProduct(product_id Int)
    - Return: fail or success

4. editProduct(product_id Int, name String, description String, category_id String, item_qty Int)
    - Return: 0 (fail) or 1 (success)

Key Risks

Skill Risks	With most of the team having little to no experience building a website of this scope. We had to use CakePhp and none of the team had used this framework before. We do have various relevant skills in specific parts of the process among our team members. As Software engineers, we are capable of learning what we need to know for the project to close our skill gaps.

Schedule Risks	With the team members having varying schedules and responsibilities that could make it difficult to meet and collaborate. We have scheduled a meeting time every Thursday as needed or online slack meeting which were split up into groups Front-end and Back-end. Occasionally, class time itself gives us a chance to meet.

Technical Risks	Security concerns over whether a user can gain admin rights by changing a bit of code to. Since PHP is rendered server side, we're feel strongly that we can restrict that functionality to only the users listed to have access in the database. Passwords shell be stored in plaintext in the database.

Teamwork Risks	There may be in the disagreements in the future as the project progresses, but we hope to resolve any issues that arise amongst ourselves. Uneven workload distribution can be possible which the team lead and tech lead will try to identify as soon as possible and resolve it.

Legal/Content Risks	Using content from third parties can cause legal issues that can compromise the project if we are careless. We will be careful to only use content that has been preapproved by our instructors, and document those that we do use. If we need something outside of the list of preapproved sources, we will explicitly ask for permission from our instructors.

The Team

Name	**Role**
Charles Williams	(**Team lead**) **QA**
Jason Huang	(**Tech lead**) **Backend**
Krishnan Ramakrishnan	**Backend**
Leopoldo Rodriguez	**Backend**
Myat Min Maung	**Frontend**
Grant Gaviglio	**Frontend**
Wilson Le	**Frontend**